Intel® Pentium® 4 Processor on 90 nm Process

Specification Update

June 2006

Notice: The Intel® Pentium® processor may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Document Number: 302352-030

Contents

§

Revision History

Revision Number	Description	Date
-001	• Initial Release	June 2004
-002	• Added content for Intel® Pentium® 4 processor on 90 nm process in 775-land package • Added 775-land package processor upside marking diagram in Figure 2 • Added processor identification information for 775-land package to Table 1 • Notes added to clarify that C0 errata only apply to 478 pin package • Modified for Processor Identification information Table Notes	"Out-of-Cycle" June 21 2004
-003	• Repaired drawings in Figures 1 and 2; reformatted document layout	"Out-of-Cycle" June 22, 2004
-004	• Separated the D0 column in *Summary Tables of Changes* into D0 and LD0 (L=LGA775) columns • Updated errata R23 in summary table of changes • Added errata R32-R38	Aug 2004
-005	• Updated Processor Identification Table, and Summary Table of Changes • Added errata R39-R54	Sept 2004
-006	• Updated Processor Identification Table, and Summary Table of Changes • Added E-stepping information • Added errata R55-R68	Out of Cycle 9/23/2004
-007	• Updated and sorted Processor Identification Table • Added errata R69-R74	October 2004
-008	• Updated Processor Identification Table • Added errata R75-R77	November 2004
-009	• Updated Processor Identification Table • Added errata R78, R79	December 2004
-010	• Updated Processor Identification Table	December 2004
-011	• Updated Processor Identification Table and Summary Table of Changes • Added errata R80, R81, and R82	January 2005

Revision Number	Description	Date
-012	• Updated Processor Identification Table and its notes and updated summary table of changes • Added errata R83, R84, R85	February 2005
-013	• Updated Processor Identification Table, and Summary Table of Changes, and processor upside marking for 660, 650, 640, and 630$^{\Delta}$ processor • Added N-stepping information • Added Errata R86	"Out of Cycle" February 22, 2005
-014	• Updated Processor Identification Table, and Summary Table of Changes • Added Errata R87, R88, R89, R90	March 2005
-015	• Updated Summary Table of Changes, and affected documents • Updated R31, R37, and added R91	April 2005
-016	• Updated R32 in Summary Table of Changes	May 2005
-017	• Updated affected documents, added/updated 5x1 and 670 part and processor upside marking, updated processor identification table,	"Out of Cycle" May 26, 2005
-018	• Updated errata R36 and R41, and added specification changes R1, and updated processor identification table	June 2005
-019	• Added erratum R92 and updated related document	July 2005
-020	• Added erratum R93, and updated processor identification table	August 2005
-021	• Added errata R94, R95, and updated processor identification table	September 2005
-022	• Added G1-stepping info, updated processor identification table and added errata R96, R97, R98	October 2005
-023	• Added R0-stepping info, updated affected document, updated processor identification table and added errata R99-R109	"Out of Cycle" November 14, 2005
-024	• Updated summary table of changes, updated processor identification table and added erratum R110	December 2005
-025	• Updated erratum R17 and added errata R111-R114, updated processor identification table, updated figure 3 and figure 4 to show Pb-free marking	January 2006
-026	• Updated related documents, updated processor identification table, added erratum R115	February 2006
-027	• Added errata R116, R117	March 2006
-028	• Added errata R118, R119, R120	April 2006
-029	• Added erratum R121	May 2006
-030	• Added erratum R122, updated processor identification table	June 2006

§

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains S-Specs, Errata, Documentation Changes, Specification Clarifications and Specification Changes.

Affected Documents

Document Title	Document Number
Intel® Pentium® 4 Processor on 90 nm Process Datasheet	300561-003 http://developer.intel.com/design/pentium4/datashts/300561.htm
Intel® Pentium® 4 Processors 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521Δ Supporting Hyper-Threading Technology Datasheet On 90 nm Process in 775-land LGA Package and *supporting Intel® Extended Memory 64 TechnologyΦ*	302351-004 http://developer.intel.com/design/pentium4/datashts/302351.htm
Intel® Pentium® 4 Processor 6xxΔ Sequence and Intel® Pentium® 4 Processor Extreme Edition Datasheet On 90 nm Process in the 775-land LGA Package and supporting Intel® Extended Memory 64 TechnologyΦ , and supporting Intel® Virtualization Technology	306382-003 http://developer.intel.com/design/pentium4/datashts/306382.htm

Related Documents

Document Title	Document Number
IA-32 Intel® Architecture Software Developer's Manual Volume 1: Basic Architecture, document 253665	http://developer.intel.com/design/pentium4/manuals/index_new.htm
IA-32 Intel® Architecture Software Developer's Manual Volume 2A: Instruction Set Reference Manual A–M, document 253666	
IA-32 Intel® Architecture Software Developer's Manual Volume 2B: Instruction Set Reference Manual, N–Z, document 253667	
IA-32 Intel Architecture Software Developer's Manual Volume 3A: System Programming Guide, document 253668	
IA-32 Intel Architecture Software Developer's Manual Volume 3B: System Programming Guide, document 253669	

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number

Errata are design defects or errors. Errata may cause the Intel® Pentium® processor's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

§

Summary Tables of Changes

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to Pentium 4 processors on 90 nm process. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Erratum, Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

Doc:	Document change or update that will be implemented.
Plan Fix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.
PKG:	This column refers to errata on the Intel® Pentium® 4 processor on 90 nm process substrate.
AP:	APIC related erratum.
Shaded:	This item is either new or modified from the previous version of the document.

Note: Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor
B = Mobile Intel® Pentium® II processor
C = Intel® Celeron® processor
D = Intel® Pentium® II Xeon® processor
E = Intel® Pentium® III processor
F = Intel® Pentium® processor Extreme Edition
G = Intel® Pentium® III Xeon® processor
H = Mobile Intel® Celeron® processor at 466 MHz, 433 MHz, 400 MHz, 366 MHz, 333 MHz, 300 MHz, and 266 MHz
K = Mobile Intel® Pentium® III Processor – M
L = Intel® Celeron® D processor
M = Mobile Intel® Celeron® processor
N = Intel® Pentium® 4 processor

O = Intel® Xeon® processor MP
P = Intel® Xeon® processor
Q = Mobile Intel® Pentium® 4 processor supporting Hyper-Threading Technology on 90-nm process technology
R = Intel® Pentium® 4 processor on 90 nm process
S = 64-bit Intel® Xeon® Processor with 800 MHz system bus
T = Mobile Intel® Pentium® 4 processor – M
U = 64-bit Intel® Xeon® processor MP with up to 8MB L3 Cache
V = Mobile Intel® Celeron® processor on 0.13 Micron Process in Micro-FCPGA Package
W = Intel® Celeron® M processor
X = Intel® Pentium® M processor on 90 nm process with 2-MB L2 cache
Y = Intel® Pentium® M processor
Z = Mobile Intel® Pentium® 4 processor with 533 MHz system bus
AA = Intel® Pentium® processor Extreme Edition and Intel® Pentium® D processor on 65nm process
AB = Intel® Pentium® 4 processor on 65 nm process
AC = Intel® Celeron® Processor in 478 Pin Package
AD = Intel® Celeron® D processor on 65 nm process
AE = Intel® Core™ Duo Processor and Intel® Core™ Solo processor on 65nm process

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R1	X	X	X	X	X	X	X	X	X	No Fix	Transaction Is Not Retried after BINIT#
R2	X	X	X	X	X	X	X	X	X	No Fix	Invalid Opcode 0FFFh Requires a ModRM Byte
R3	X	X	X	X	X	X	X	X	X	No Fix	Processor May Hang Due to Speculative Page Walks to Non-Existent System Memory
R4	X	X	X	X	X	X	X	X	X	No Fix	Memory Type of the Load Lock Different from Its Corresponding Store Unlock
R5	X	X	X	X	X	X	X	X	X	No Fix	Machine Check Architecture Error Reporting and Recovery May Not Work As Expected
R6	X	X	X	X	X	X	X	X	X	No Fix	Debug Mechanisms May Not Function as Expected
R7	X	X	X	X	X	X	X	X	X	No Fix	Cascading of Performance Counters Does Not Work Correctly When Forced Overflow Is Enabled
R8	X	X	X	X	X	X	X	X	X	No Fix	EMON Event Counting of x87 Loads May Not Work As Expected
R9	X	X	X	X	X	X	X	X	X	No Fix	System Bus Interrupt Messages without Data Which Receive a HardFailure Response May Hang the Processor

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R10	X	X	X	X	X	X	X	X	X	No Fix	The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction
R11	X	X	X	X	X	X	X	X	X	No Fix	FSW May Not Be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions
R12	X	X	X	X	X	X	X	X	X	No Fix	Processor Issues Inconsistent Transaction Size Attributes for Locked Operation
R13	X	X	X	X	X	X	X	X	X	No Fix	When the Processor Is in the System Management Mode (SMM), Debug Registers May Be Fully Writeable
R14	X	X	X	X	X	X	X	X	X	No Fix	Shutdown and IERR# May Result Due to a Machine Check Exception on a Hyper-Threading Technology Enabled Processor
R15	X	X	X	X	X	X	X	X	X	No Fix	Processor May Hang under Certain Frequencies and 12.5% STPCLK# Duty Cycle
R16	X	X	X	X	X	X	X	X	X	No Fix	System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL}
R17	X	X	X	X	X	X	X	X	X	No Fix	A Write to an APIC Registers Sometimes May Appear to Have Not Occurred
R18	X									Fixed	Some Front Side Bus I/O Specifications are not Met
R19	X	X	X	X	X	X	X	X	X	No Fix	Parity Error in the L1 Cache May Cause the Processor to Hang
R20	X									Fixed	BPM4# Signal Not Being Asserted According to Specification
R21	X	X	X	X	X					Fixed	Sequence of Locked Operations Can Cause Two Threads to Receive Stale Data and Cause Application Hang
R22	X	X	X							Fixed	A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to be Reported to the #GP Exception Handler
R23	X	X	X	X	X	X	X	X	X	No Fix	Bus Locks and SMC Detection May Cause the Processor to Hang Temporarily

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R24	X									Fixed	PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified
R25	X	X	X							Fixed	Incorrect Physical Address Size Returned by CPUID Instruction
R26	X	X	X	X	X	X	X	X	X	No Fix	Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint is set on an FP Instruction
R27	X	X	X	X	X	X	X	X	X	No Fix	xAPIC May Not Report Some Illegal Vector Errors
R28	X	X	X	X	X					Fixed	Enabling No-Eviction Mode (NEM) May Prevent the Operation of the Second Logical Processor in a Hyper-Threading Technology Enabled Processor
R29	X	X	X	X	X	X	X	X	X	No Fix	Incorrect Duty Cycle is Chosen when On-Demand Clock Modulation is Enabled in a Processor Supporting Hyper-Threading Technology
R30	X	X	X	X	X	X	X	X	X	No Fix	Memory Aliasing of Pages as Uncacheable Memory Type and Write Back (WB) May Hang the System
R31	X	X	X	X	X	X	X	X	X	No Fix	Interactions Between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior
R32	X									Fixed	STPCLK# Signal Assertion under Certain Conditions May Cause a System Hang
R33	X									Fixed	Missing Stop Grant Acknowledge Special Bus Cycle May Cause a System Hang
R34	X									Fixed	Changes to CR3 Register do not Fence Pending Instruction Page Walks
R35	X									Fixed	Simultaneous Page Faults at Similar Page Offsets on Both Logical Processors of a Hyper-Threading Technology Enabled Processor May Cause Application Failure
R36	X									Fixed	The State of the Resume Flag (RF Flag) in a Task-State Segment (TSS) May be Incorrect
R37	X	X	X	X	X	X	X	X	X	No Fix	Using STPCLK# and Executing Code From Very Slow Memory Could Lead to a System Hang

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R38	X	X	X	X	X	X	X	X	X	No Fix	Processor Provides a 4-Byte Store Unlock After an 8-Byte Load Lock
R39	X	X	X	X	X	X	X	X	X	No Fix	Data Breakpoints on the High Half of a Floating Point Line Split may not be Captured
R40	X									Fixed	CPUID Instruction May Report Incorrect L2 Associativity in Leaf 0x80000006
R41	X									Fixed	The FP_ASSIST EMON Event May Return an Incorrect Count
R42	X	X	X	X	X	X	X	X	X	No Fix	Machine Check Exceptions May not Update Last-Exception Record MSRs (LERs)
R43	X	X	X	X	X	X	X	X	X	No Fix	MOV CR3 Performs Incorrect Reserved Bit Checking When in PAE Paging
R44	X	X	X	X	X	X	X	X	X	No Fix	Stores to Page Tables May Not Be Visible to Pagewalks for Subsequent Loads Without Serializing or Invalidating the Page Table Entry
R45		X	X	X	X	X	X	X		Fixed	Execution of IRET or INTn Instructions May Cause Unexpected System Behavior
R46	X	X	X							Fixed	A Split Store Memory Access May Miss a Data Breakpoint
R47		X	X							Fixed	EFLAGS.RF May be Incorrectly Set After an IRET Instruction
R48	X									Fixed	Read for Ownership and Simultaneous Fetch May Cause the Processor to Hang
R49		X	X							Fixed	Writing the Echo TPR Disable Bit in IA32_MISC_ENABLE May Cause a #GP Fault
R50	X									Fixed	Cache Lock with Simultaneous Invalidate external snoop and SMC check May Cause the Processor to Hang
R51	X									Fixed	IRET Instruction Performing Task Switch May Not Serialize the Processor Execution
R52	X	X	X							Fixed	Incorrect Access Controls to MSR_LASTBRANCH_0_FROM_LIP MSR Registers
R53		X	X	X	X					Fixed	Recursive Page Walks May Cause a System Hang

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R54	X5	X	X							Fixed	WRMSR to bit[0] of IA32_MISC_ENABLE Register Changes Only One Logical Processor on a Hyper-Threading Technology Enabled Processor
R55			X5		X5		X	X		Fixed	VERR/VERW Instructions May Cause #GP Fault when Descriptor is in Non-canonical Space
R56			X		X					Fixed	The Base of a Null Segment May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)Φ
R57			X		X					Fixed	Upper 32 Bits of FS/GS with Null Base May not get Cleared in Virtual-8086 Mode on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled
R58			X		X		X	X	X	No Fix	Processor May Fault when the Upper 8 Bytes of Segment Selector is Loaded From a Far Jump Through a Call Gate via the Local Descriptor Table
R59			X		X		X	X	X	No Fix	Loading a Stack Segment with a Selector that References a Non-canonical Address can Lead to a #SS Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R60			X		X		X	X	X	No Fix	FXRSTOR May Not Restore Non-canonical Effective Addresses on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled
R61			X		X		X	X	X	No Fix	A Push of ESP that Faults may Zero the Upper 32 Bits of RSP
R62					X					Fixed	Enhanced Halt State (C1E) Voltage Transition May Affect a System's Power Management in a Hyper-Threading Technology Enabled Processor
R63					X		X	X	X	No Fix	Enhanced Halt State (C1E) May Not Be Entered in a Hyper-Threading Technology Enabled Processor
R64					X					Fixed	When the Execute Disable Bit Function is Enabled a Page-fault in a Mispredicted Branch May Result in a Page-fault Exception

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R65					X					Fixed	Execute Disable Bit Set with AD Assist Will Cause Livelock
R66					X					Fixed	The Execute Disable Bit Fault May be Reported Before Other Types of Page Fault When Both Occur
R67					X					Fixed	Writes to IA32_MISC_ENABLE May Not Update Flags for Both Logical Processors Threads
R68					X					Fixed	Execute Disable Mode Bit Set with CR4.PAE May Cause Livelock
R69	X	X	X	X	X	X	X	X	X	No Fix	Checking of Page Table Base Address May Not Match the Address Bit Width Supported by the Platform
R70	X	X	X	X	X	X	X	X	X	No Fix	The IA32_MCi_STATUS MSR May Improperly Indicate that Additional MCA Information May Have Been Captured
R71	X									Fixed	Execution of an Instruction with a Code Breakpoint Inhibited by the RF (Resume Flag) Bit May be Delayed by an RFO (Request For Ownership) from Another Bus Agent
R72	X	X	X	X	X	X	X	X	X	No Fix	With TF (Trap Flag) Asserted, FP Instruction That Triggers an Unmasked FP Exception May Take Single Step Trap Before Retirement of Instruction
R73	X	X	X							Fixed	MCA Corrected Memory Hierarchy Error Counter May Not Increment Correctly
R74	X	X	X	X	X	X	X	X	X	No Fix	BTS(Branch Trace Store) and PEBS(Precise Event Based Sampling) May Update Memory outside the BTS/PEBS Buffer
R75			X		X		X	X		Fixed	The Base of an LDT (Local Descriptor Table) Register May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R76					X		X	X		Fixed	L-bit of the CS and LMA bit of the IA32_EFER Register May Have an Erroneous Value For One Instruction Following a Mode Transition in a Hyper-Threading Enabled Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T).

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R77	X	X	X	X	X	X	X	X	X	No Fix	Memory Ordering Failure May Occur with Snoop Filtering Third Party Agents after Issuing and Completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) Transaction
R78	X	X	X	X	X	X	X	X	X	No Fix	Control Register 2 (CR2) Can be Updated during a REP MOVS/STOS Instruction with Fast Strings Enabled
R79			X		X		X	X		Fixed	TPR (Task Priority Register) Updates during Voltage Transitions of Power Management Events May Cause a System Hang
R80			X		X		X	X	X	No Fix	REP STOS/MOVS Instructions with RCX >=2^32 May Cause a System Hang
R81			X		X		X	X		Fixed	An REP MOVS or an REP STOS Instruction with RCX >= 2^32 May Fail to Execute to Completion or May Write to Incorrect Memory Locations on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R82			X		X		X	X		Fixed	An REP LODSB or an REP LODSD or an REP LODSQ Instruction with RCX >= 2^32 May Cause a System Hang on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R83			X		X		X	X		No Fix	A Data Access which Spans Both the Canonical and the Non-Canonical Address Space May Hang the System
R84	X	X	X	X	X	X	X	X		Fixed	Running in SMM (System Management Mode) And L1 Data Cache Adaptive Mode May Cause Unexpected System Behavior when SMRAM is Mapped to Cacheable Memory
R85			X							Fixed	CPUID Instruction Incorrectly Reports CMPXCH16B as Supported
R86				X	X	X	X	X		Fixed	Unaligned PDPTR (Page-Directory-Pointer) Base with 32-bit Mode PAE (Page Address Extension) Paging May Cause Processor to Hang

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R87			X							Fixed	FXSAVE Instruction May Result in Incorrect Data on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R88			X							Fixed	Compatibility Mode STOS Instructions May Alter RSI Register Results on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R89			X							Fixed	LDT Descriptor Which Crosses 16 bit Boundary Access Does Not Cause a #GP Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R90			X							Fixed	Upper Reserved Bits are Incorrectly Checked While Loading PDPTR's on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)
R91			X		X		X	X	X	No Fix	A 64-Bit Value of Linear Instruction Pointer (LIP) May be Reported Incorrectly in the Branch Trace Store (BTS) Memory Record or in the Precise Event Based Sampling (PEBS) Memory Record
R92	X	X	X	X	X	X	X	X		Fixed	It is Possible That Two specific Invalid Opcodes May Cause Unexpected Memory Accesses
R93	X	X	X	X	X	X	X	X	X	No Fix	At Core-to-bus Ratios of 16:1 and Above Defer Reply Transactions with Non-zero REQb Values May Cause a Front Side Bus Stall
R94	X	X	X	X	X	X	X	X	X	No Fix	The Processor May Issue Front Side Bus Transactions up to 6 Clocks after RESET# is Asserted
R95	X	X	X	X	X	X	X	X	X	No Fix	Front Side Bus Machine Checks May be Reported as a Result of On-Going Transactions during Warm Reset
R96							X			Fixed	CPUID Feature Flag Reports LAHF/SAHF as Unavailable however the Execution of LAHF/SAHF May Not Result in an Invalid Opcode Exception
R97	X5	X5	X5	X5	X5	X5	X5	X5	X	No Fix	The Processor May Issue Multiple Code Fetches to the Same Cache Line for Systems with Slow Memory

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R98	X	X	X	X	X	X	X	X	X	No Fix	Writing the Local Vector Table (LVT) when an Interrupt is Pending May Cause an Unexpected Interrupt
R99									X	No Fix	Access to an Unsupported Address Range in Uniprocessor (UP) or Dual-processor (DP) Systems Supporting Intel® Virtualization Technology May Not Trigger Appropriate Actions
R100									X	No Fix	VM Exit Due to a MOV from CR8 May Cause an Unexpected Memory Access
R101									X	No Fix	The Processor May Incorrectly Respond to Machine Checks during VM Entry/Exit Transitions
R102									X	No Fix	INIT during String Operations in the Virtual-Machine Extension (VMX) Guest Mode May Cause Unexpected System Behavior
R103									X	No Fix	Power Down Requests May not be Serviced if a Power Down Transition is Interrupted by an In-Target Probe Event in the Presence of a Specific Type of VM Exit
R104									X	No Fix	VM EXIT Due to TPR shadow Below Threshold May Improperly Set and Cause "Blocking by STI" actions
R105									X	No Fix	VM Entry/Exit Writes to LSTAR/SYSCALL_FLAG MSR's May Cause Incorrect Data to be Written to Bits [63:32]
R106									X	No Fix	Machine Check Architecture Multiple Data Parity Errors May be Reported
R107									X	Plan Fix	Attempting to Use an LDT Entry when the LDTR Has Been Loaded with an Unusable Segment May Cause Unexpected Memory Accesses
R108									X	No Fix	The Execution of a VMPTRLD Instruction May Cause an Unexpected Memory Access
R109									X	No Fix	The Execution of VMPTRLD or VMREAD May Cause an Unexpected Memory Access
R110			X		X		X	X	X	No Fix	IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception

NO.	C0[1]	D0	LD0[2]	E0	LE0[2]	G1[1]	LG1[2]	LN0[2]	LR0[2]	Plan	ERRATA
R111									X	No Fix	FS/GS Base MSRs can be Loaded from MSR-Load Areas during VM Entry or VM Exit
R112									X	Plan Fix	NMI-blocking Information Recorded in VMCS May be Incorrect after a #GP on an IRET Instruction
R113									X	Plan Fix	VMLAUNCH/VMRESUME May Not Fail when VMCS is Programmed to Cause VM Exit to Return to a Different Mode
R114			X							Fixed	Upper 32 bits of 'From' Address Reported through LBR or LER MSRs, BTMs or BTSs May be Incorrect
R115									X	Plan Fix	VMEntry from 64-bit Host to 32-bit Guest may Cause IERR# with Hyper-Threading Enabled
R116	X	X	X	X	X	X	X	X	X	No Fix	L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion
R117									X	Plan Fix	VMExit after MOV SS and a Waiting x87 Instruction May not Clear the Interruptibility State in the VMM's Working VMCS
R118									X	Plan Fix	VMCALL to Activate Dual-monitor Treatment of SMIs and SMM Ignores Reserved Bit settings in VM-exit Control Field
R119	X	X	X	X	X	X	X	X	X	No Fix	Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations
R120	X	X	X	X	X	X	X	X	X	No Fix	Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue
R121	X	X	X	X	X	X	X	X	X	No Fix	The IA32_MC0_STATUS and IA32_MC1_STATUS Overflow Bit is not set when Multiple Un-correctable Machine Check Errors Occur at the Same Time
R122	X	X	X	X	X	X	X	X	X	No Fix	Debug Status Register (DR6) Breakpoint Condition Detected Flags May be set Incorrectly

NOTES:
1. Only applies to Pentium® 4 processor on 90 nm Process in the 478-pin package
2. Prefix "L" denotes Pentium 4 processor on 90 nm Process in the 775-land LGA package
3. This erratum applies to Pentium 4 processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T) for Single-Processor Server/Workstation Platform configurations only. Non-server/workstation desktop configurations do not support the Intel Extended Memory 64 Technology.
4. This erratum does not apply to Pentium 4 processors for single-processor server/workstation platform configurations.
5. For these steppings, this erratum may be worked around in BIOS.

NO.	SPECIFICATION CHANGES
R1	Land Assignment Specification Change

NO.	SPECIFICATION CLARIFICATIONS
	There are no Specification Clarification in this Specification Update revision

NO.	DOCUMENTATION CHANGES
	There are no documentation changes in this Specification Update revision

§

General Information

Figure 1. Intel® Pentium® 4 Processor on 90 nm Process in the 478-pin Package

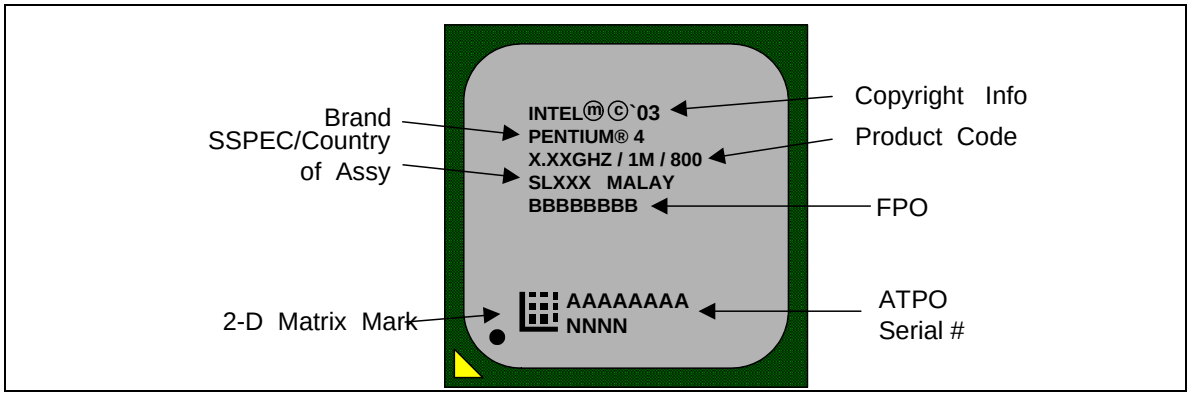


Figure 2. Intel® Pentium® 4 Processors 570, 560, 550, 540, 530 and 520 Δ Supporting Hyper-Threading Technology on 90 nm Process in the 775-Land LGA Package

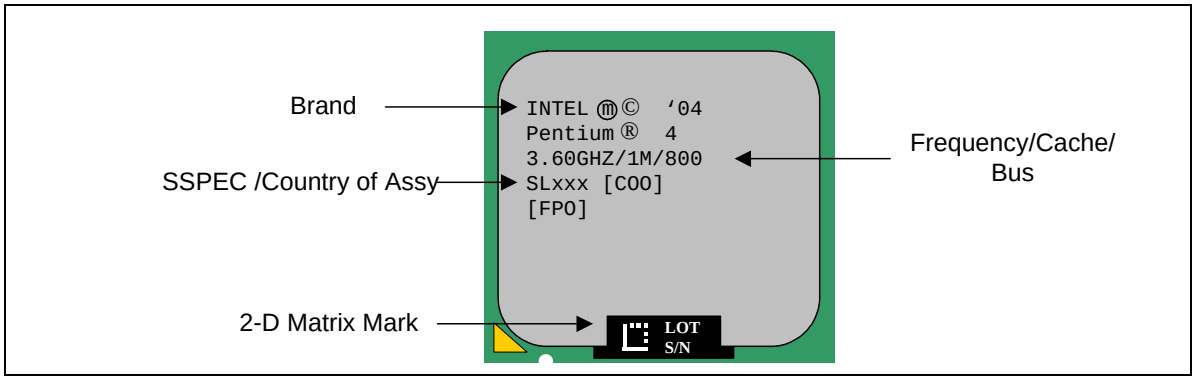


Figure 3. Intel® Pentium® 4 Processors 571, 561, 551, 541, 531 and 521Δ Supporting Hyper-Threading Technology on 90 nm Process in the 775-Land LGA Package

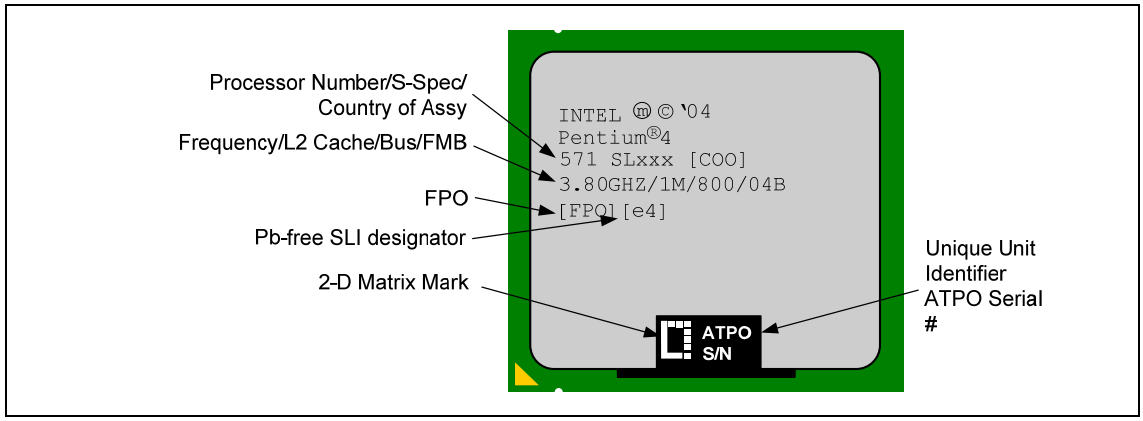


Figure 4. Intel® Pentium® 4 Processor 670, 660, 650, 640, and 630Δ on 90 nm Process in the 775-Land LGA Package

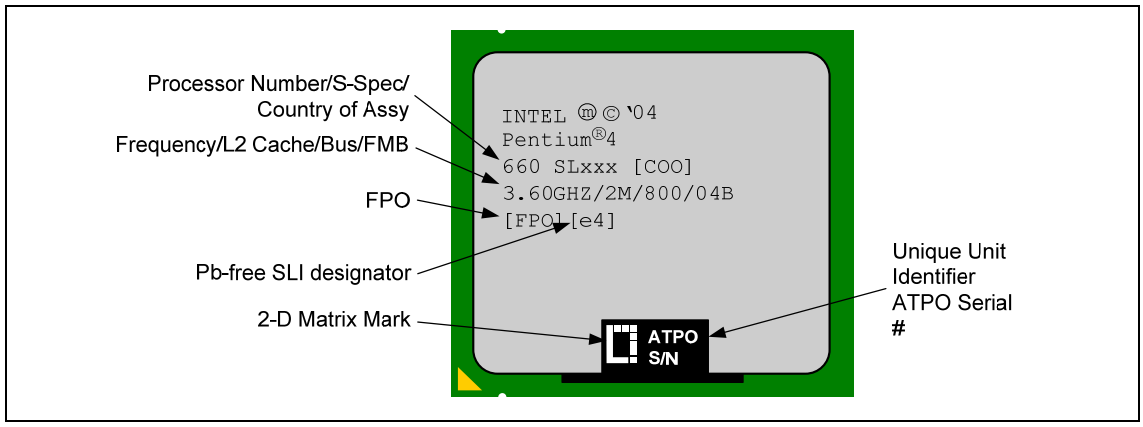
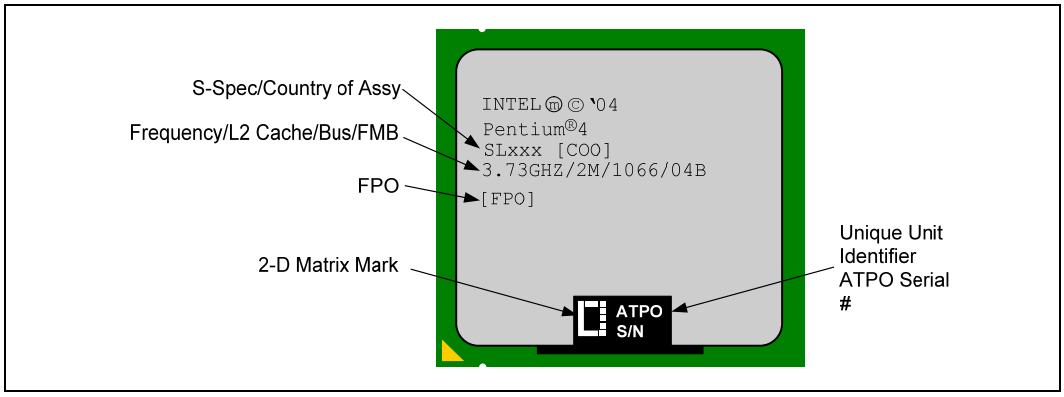


Figure 5. Intel® Pentium® 4 Processor Extreme Edition on 90 nm Process in the 775-Land LGA Package



§

Identification Information

The Pentium 4 processor on 90 nm process can be identified by the following values:

Family[1]	Model[2]
1111b	0011b
1111b	0100b

NOTES:
1. The Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL7D7	C0	512K	0F33h	2.26GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 19
SL7FY	C0	1M	0F33h	2.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL7E8	C0	1M	0F33h	2.40GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7
SL7E9	C0	1M	0F33h	2.66GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 19
SL7D8	C0	1M	0F33h	2.80GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7
SL79K	C0	1M	0F33h	2.80GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL79L	C0	1M	0F33h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL79M	C0	1M	0F33h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 6, 11
SL7B8	C0	1M	0F33h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 6, 11
SL7B9	C0	1M	0F33h	3.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	4, 6, 11
SL7AJ	C0	1M	0F33h	3.40GHz//800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 6, 11

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL7E2	D0	1M	0F34h	2.80GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7
SL7E3	D0	1M	0F34h	2.80GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL7KA	D0	1M	0F34h	2.80GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7K9	D0	1M	0F34h	2.80GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7
SL7E4	D0	1M	0F34h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL7KB	D0	1M	0F34h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7L4	D0	1M	0F34h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7L5	D0	1M	0F34h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7E5	D0	1M	0F34h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL7KC	D0	1M	0F34h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7E6	D0	1M	0F34h	3.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 6, 11
SL7KD	D0	1M	0F34h	3.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 6, 11
SL7YP	D0	1M	0F34h	2.40GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7
SL7YU	D0	1M	0F34	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 19
SL7J4	D0	1M	0F34h	2.80GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8
SL7J5	D0	1M	0F34h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11
SL7KH	D0	1M	0F34h	2.80GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8
SL7KJ	D0	1M	0F34h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11
SL7YV	D0	1M	0F34	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 19
SL7J6	D0	1M	0F34h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11
SL7KK	D0	1M	0F34h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL7J7	D0	1M	0F34h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11
SL7KL	D0	1M	0F34h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11
SL7LA	D0	1M	0F34h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12
SL7J8	D0	1M	0F34h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 9, 11
SL7KM	D0	1M	0F34h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 9, 11
SL7L8	D0	1M	0F34h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12
SL7J9	D0	1M	0F34h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 9, 11, 13
SL7KN	D0	1M	0F34h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 9, 11, 13
SL7L9	D0	1M	0F34h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 13
SL88F	E0	1M	0F41h	2.40GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7
SL8B3	E0	1M	0F41h	2.66GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 19
SL88G	E0	1M	0F41h	2.80GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7
SL88H	E0	1M	0F41h	2.80GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7PL	E0	1M	0F41h	2.80GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL7PK	E0	1M	0F41h	2.80GHz/533MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7
SL7PM	E0	1M	0F41h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL88J	E0	1M	0F41h	3.00GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL7PN	E0	1M	0F41h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL88K	E0	1M	0F41h	3.20GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 4, 7, 11
SL88L	E0	1M	0F41h	3.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	1, 2, 4, 7, 11
SL7PP	E0	1M	0F41h	3.40GHz/800MHz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL7PT	E0	1M	0F41h	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 15, 19
SL82V	E0	1M	0F41h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 15
SL7PR	E0	1M	0F41h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 15
SL8HX	E0	1M	0F41h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 15
SL85U	E0	1M	0F41h	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 12, 15, 19
SL8U5	E0	1M	0F41h	2.80GHz/533Mhz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	3, 4, 8, 12, 15, 19
SL8J8	E0	1M	0F41h	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 12, 15, 19
SL85V	E0	1M	0F41h	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 13, 14, 15, 19
SL8J9	E0	1M	0F41h	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 12, 13, 14, 15, 19
SL87L	E0	1M	0F41h	3.06GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 13, 14, 15, 19
SL8JA	E0	1M	0F41h	3.06GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 12, 13, 14, 15, 19
SL82X	E0	1M	0F41h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11, 14, 15
SL7PU	E0	1M	0F41h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 14, 15
SL8HZ	E0	1M	0F41h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL7PW	E0	1M	0F41h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 14, 15
SL7PX	E0	1M	0F41h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL82Z	E0	1M	0F41h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11, 14, 15
SL8J2	E0	1M	0F41h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL7PY	E0	1M	0F41h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 14, 15
SL7PZ	E0	1M	0F41h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL833	E0	1M	0F41h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11, 14, 15
SL7ZW	E0	1M	0F41h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11, 14, 15, 16

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL8J5	E0	1M	0F41h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL84X	E0	1M	0F41h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 9, 11, 13, 14, 15
SL7Q2	E0	1M	0F41h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 13, 14, 15
SL7NZ	E0	1M	0F41h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 13, 14, 15
SL8J6	E0	1M	0F41h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 13, 14, 15
SL82U	E0	1M	0F41h	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 13, 14, 15
SL84Y	E0	1M	0F41h	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 9, 11, 13, 14, 15
SL7P2	E0	1M	0F41h	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 13, 14, 15
SL8J7	E0	1M	0F41h	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 13, 14, 15
SL8K4	G1	1M	0F49h	3.40GHz/800Mhz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL8K2	G1	1M	0F49h	3.20GHz/800Mhz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL8JZ	G1	1M	0F49h	3.00GHz/800Mhz	35.0 x 35.0 mm FC-mPGA4 Rev 2.0	2, 4, 7, 11
SL8PL	G1	1M	0F49h	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 12, 15, 19
SL9CK	G1	1M	0F49h	2.66GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 12, 15, 19
SL8U4	G1	1M	0F49h	2.80GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 12, 15, 19
SL9CJ	G1	1M	0F49h	2.80GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 12, 15, 19
SL8PP	G1	1M	0F49h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 15
SL9CG	G1	1M	0F49h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 15
SL8JX	G1	1M	0F49h	2.80GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 12, 15, 19
SL9CD	G1	1M	0F49h	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	1, 4, 8, 11, 12, 15, 19
SL8PM	G1	1M	0F49h	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 12, 13, 14, 15, 19
SL8ZY	G1	1M	0F49h	2.93GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11, 12, 13, 14, 15, 19

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL8ZZ	G1	1M	0F49h	3.06GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11, 12, 13, 14, 15, 19
SL8PN	G1	1M	0F49h	3.06GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 12, 13, 14, 15, 19
SL9CA	G1	1M	0F49h	3.06GHz/533MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 8, 11, 12, 15, 19
SL8PQ	G1	1M	0F49h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL9CB	G1	1M	0F49h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL8PR	G1	1M	0F49h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL9C6	G1	1M	0F49h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL8PS	G1	1M	0F49h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL9C5	G1	1M	0F49h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 14, 15
SL8AB	N0	2M	0F43h	2.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 8, 11, 12, 15, 19
SL7Z9	N0	2M	0F43h	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 5, 8, 11, 12, 14, 15, 16
SL7Z8	N0	2M	0F43h	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 5, 8, 11, 12, 14, 15, 16
SL7Z7	N0	2M	0F43h	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 5, 8, 11, 12, 14, 15, 16
SL7Z5	N0	2M	0F43h	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 5, 9, 11, 12, 13, 14, 15, 16
SL7Z4	N0	2M	0F43h	3.73GHz/1066MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 9, 11, 12, 15
SL7Z3	N0	2M	0F43h	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	2, 4, 5, 9, 11, 12, 13, 14, 15, 16
SL8Q7	R0	2M	0F4Ah	3.00GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 8, 11, 12, 14, 15, 16, 18
SL8Q6	R0	2M	0F4Ah	3.20GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 8, 11, 12, 14, 15, 16, 18
SL8Q5	R0	2M	0F4Ah	3.40GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 8, 11, 12, 14, 15, 16, 18
SL8UP	R0	2M	0F4Ah	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19
SL8PZ	R0	2M	0F4Ah	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 9, 11, 12, 13, 14, 15, 16, 18

Table 1. Intel® Pentium® 4 Processor on 90 nm Process Processor Identification Information

S-Spec	Core Stepping	L2 Cache Size (bytes)	CPUID	Speed Core/Bus	Package and Revision	Notes
SL8QB	R0	2M	0F4Ah	3.60GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 9, 11, 12, 13, 14, 15, 16, 17, 18
SL8PY	R0	2M	0F4Ah	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 9, 11, 12, 13, 14, 15, 16, 18
SL8Q9	R0	2M	0F4Ah	3.80GHz/800MHz	775-land FC-LGA4 37.5 x 37.5 mm Rev 01	4, 5, 9, 11, 12, 13, 14, 15, 16, 17, 18

NOTES:
1. This is a boxed Intel Pentium 4 processor with an unattached fan heatsink.
2. Some of these processors are offered as boxed processors with an unattached fan heatsink.
3. These are engineering samples only.
4. These parts are multiple VIDs.
5. These parts will only operate at the specified core to bus frequency ratio and lower.
6. These Pentium 4 processors on 90 nm process support loadline A (FMB1.5).
7. These Pentium 4 processors on 90 nm process support loadline B (FMB1.0).
8. These Pentium 4 processors on 90 nm process in 775-land LGA package support the 775_VR_CONFIG_04A (mainstream) specifications.
9. These Pentium 4 processors on 90 nm process in 775-land LGA package support the 775_VR_CONFIG_04B (performance) specifications.
10. These parts have following specifications: VID = 1.475, Vmax = 1.370 V, Vmin = 1.290 V, VID = 1.500, Vmax = 1.395 V, Vmin = 1.315 V and VID = 1.525, Vmax = 1.420 V, Vmin = 1.340 V, Icc_max = 55.9 A, TDP = 68.4 W, Tcase = 75 °C, Isgnt = 23.0 A.
11. These parts support Hyper-Threading Technology.
12. These parts support Intel® Extended Memory 64 Technology.
13. These parts support Thermal Monitor 2 feature.
14. These parts support Enhanced Halt State.
15. These parts support Execute Disable Bit Feature.
16. These parts support Enhanced Intel SpeedStep® technology.
17. These parts support Intel® Virtualization Technology
18. These parts support minimum bus ratio of 12:1
19. This SKU was produced in limited quantity and is no longer available for purchase

§

Errata

R1. Transaction Is Not Retried after BINIT#

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, it will not be retried.

Implication: When this erratum occurs, locked transactions will unexpectedly not be retried.

Workaround: None identified.

Status: For the steppings affected see the *Summary Tables of Changes*.

R2. Invalid Opcode 0FFFh Requires a ModRM Byte

Problem: Some invalid opcodes require a ModRM byte (or other following bytes), while others do not. The invalid opcode 0FFFh did not require a ModRM byte in previous generation Intel architecture processors, but does in the Pentium 4 processor.

Implication: The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Pentium 4 processor.

Workaround: Use a ModRM byte with invalid 0FFFh opcode.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R3. Processor May Hang Due to Speculative Page Walks to Non-Existent System Memory

Problem: A load operation that misses the Data Translation Lookaside Buffer (DTLB) will result in a page-walk. If the page-walk loads the Page Directory Entry (PDE) from cacheable memory and that PDE load returns data that points to a valid Page Table Entry (PTE) in uncacheable memory the processor will access the address referenced by the PTE. If the address referenced does not exist the processor will hang with no response from system memory.

Implication: Processor may hang due to speculative page walks to non-existent system memory.

Workaround: Page directories and page tables in UC memory space which are marked valid must point to physical addresses that will return a data response to the processor.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R4. Memory Type of the Load Lock Different from Its Corresponding Store Unlock

Problem: A use-once protocol is employed to ensure that the processor in a multi-agent system may access data that is loaded into its cache on a Read-for-Ownership operation at least once before it is snooped out by another agent. This protocol is necessary to avoid a multi-agent livelock scenario in which the processor cannot gain ownership of a line and modify it before that data is snooped out by another agent. In the case of this erratum, split load lock instructions incorrectly trigger the use-once protocol. A load lock operation accesses data that splits across a page boundary with both pages of WB memory type. The use-once protocol activates and the memory type for the split halves get forced to UC. Since use-once does not apply to stores, the store unlock instructions go out as WB memory type. The full sequence on the bus is: locked partial read (UC), partial read (UC), partial write (WB), locked partial write (WB). The use-once protocol should not be applied to load locks.

Implication: When this erratum occurs, the memory type of the load lock will be different than the memory type of the store unlock operation. This behavior (load locks and store unlocks having different memory types) does not introduce any functional failures such as system hangs or memory corruption.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R5. Machine Check Architecture Error Reporting and Recovery May Not Work As Expected

Problem: When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

- When a transaction is deferred during the snoop phase and subsequently receives a Hard Failure response, the transaction should be removed from the bus queue so that the processor may proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.
- When a hardware prefetch results in an uncorrectable tag error in the L2 cache, MC0_STATUS.UNCOR and MC0_STATUS.PCC are set but no Machine Check Exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE, .the information for that error will not be logged because MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.
- When the reporting of errors is disabled for Machine Check Architecture (MCA) Bank 2 by setting all MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.

- When one-half of a 64-byte instruction fetch from the L2 cache has an uncorrectable error and the other 32-byte half of the same fetch from the L2 cache has a correctable error, the processor will attempt to correct the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.
- When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR REGISTER (MC1_ADDR). In some instances of a parity error on a load operation that hits the L1 cache, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.
- When an error exists in the tag field of a cache line such that a request for ownership (RFO) issued by the processor hits multiple tag fields in the L2 cache (the correct tag and the tag with the error) and the accessed data also has a correctable error, the processor will correctly log the multiple tag match error but will hang when attempting to execute the machine check exception handler.
- If a memory access receives a machine check error on both 64 byte halves of a 128-byte L2 cache sector, the IA32_MC0_STATUS register records this event as multiple errors, i.e., the valid error bit and the overflow error bit are both set indicating that a machine check error occurred while the results of a previous error were in the error-reporting bank. The IA32_MC1_STATUS register should also record this event as multiple errors but instead records this event as only one correctable error.
- The overflow bit should be set to indicate when more than one error has occurred. The overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA overflow bit will not be updated, thereby incorrectly indicating only one error has been received.
- If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a Machine Check Exception (MCE). If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, while attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is successfully completed, it will attempt to restart the I/O instruction, but will not have the correct machine state due to the call to the MCE handler. This can lead to failure of the restart and shutdown of the processor.
- If PWRGOOD is de-asserted during a RESET# assertion causing internal glitches, the MCA registers may latch invalid information.
- If RESET# is asserted, then de-asserted, and reasserted, before the processor has cleared the MCA registers, then the information in the MCA registers may not be reliable, regardless of the state or state transitions of PWRGOOD.
- If MCERR# is asserted by one processor and observed by another processor, the observing processor does not log the assertion of MCERR#. The Machine Check Exception (MCE) handler called upon assertion of MCERR# will not have any way to determine the cause of the MCE.

- The Overflow Error bit (bit 62) in the IA32_MC0_STATUS register indicates, when set, that a machine check error occurred while the results of a previous error were still in the error reporting bank (i.e. The Valid bit was set when the new error occurred). If an uncorrectable error is logged in the error-reporting bank and another error occurs, the overflow bit will not be set.
- The MCA Error Code field of the IA32_MC0_STATUS register gets written by a different mechanism than the rest of the register. For uncorrectable errors, the other fields in the IA32_MC0_STATUS register are only updated by the first error. Any further errors that are detected will update the MCA Error Code field without updating the rest of the register, thereby leaving the IA32_MC0_STATUS register with stale information.
- When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_Status Register may be updated with incorrect information. The IA32_MC1_Status Register should not be updated for speculative loads.
- The processor should only log the address for L1 parity errors in the IA32_MC1_Status register if a valid address is available. If a valid address is not available, the Address Valid bit in the IA32_MC1_Status register should not be set. In instances where an L1 parity error occurs and the address is not available because the linear to physical address translation is not complete or an internal resource conflict has occurred, the Address Valid bit is incorrectly set.
- The processor may hang when an instruction code fetch receives a hard failure response from the system bus. This occurs because the bus control logic does not return data to the core, leaving the processor empty. IA32_MC0_STATUS MSR does indicate that a hard fail response occurred.
- The processor may hang when the following events occur and the machine check exception is enabled, CR4.MCE=1. A processor that has it's STPCLK# pin asserted will internally enter the Stop Grant State and finally issue a Stop Grant Acknowledge special cycle to the bus. If an uncorrectable error is generated during the Stop Grant process it is possible for the Stop Grant special cycle to be issued to the bus before the processor vectors to the machine check handler. Once the chipset receives its last Stop Grant special cycle it is allowed to ignore any bus activity from the processors. As a result, processor accesses to the machine check handler may not be acknowledged, resulting in a processor hang.

Implication: The processor is unable to correctly report and/or recover from certain errors.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R6. Debug Mechanisms May Not Function As Expected

Problem: Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

- When the following conditions occur: 1) An FLD instruction signals a stack overflow or underflow, 2) the FLD instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a Debug Register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.
- When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating point store using the Extended Real data type, and an unmasked floating point exception occurs on the store, the break point will not be captured.
- When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a REP instruction. As an example, during execution of a REP MOVSW instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.
- A data breakpoint that is set on a load to uncacheable memory may be ignored due to an internal segment register access conflict. In this case the system will continue to execute instructions, bypassing the intended breakpoint. Avoiding having instructions that access segment descriptor registers, e.g., LGDT, LIDT close to the UC load, and avoiding serialized instructions before the UC load will reduce the occurrence of this erratum.

Implication: Certain debug mechanisms do not function as expected on the processor.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R7. Cascading of Performance Counters Does Not Work Correctly When Forced Overflow Is Enabled

Problem: The performance counters are organized into pairs. When the CASCADE bit of the Counter Configuration Control Register (CCCR) is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

Implication: The performance counters do not cascade when the FORCE_OVF bit is set.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R8. EMON Event Counting of x87 Loads May Not Work As Expected

Problem: If a performance counter is set to count x87 loads and floating point exceptions are unmasked, the FPU Operand Data Pointer (FDP) may become corrupted.

Implication: When this erratum occurs, the FPU Operand Data Pointer (FDP) may become corrupted.

Workaround: This erratum will not occur with floating point exceptions masked. If floating point exceptions are unmasked, then performance counting of x87 loads should be disabled.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R9. System Bus Interrupt Messages without Data Which Receive a HardFailure Response May Hang the Processor

Problem: When a system bus agent (processor or chipset) issues an interrupt transaction without data onto the system bus and the transaction receives a HardFailure response, a potential processor hang can occur. The processor, which generates an inter-processor interrupt (IPI) that receives the HardFailure response, will still log the MCA error event cause as HardFailure, even if the APIC causes a hang. Other processors, which are true targets of the IPI, will also hang on hardfail-without-data, but will not record an MCA HardFailure event as the cause. If a HardFailure response occurs on a system bus interrupt message with data, the APIC will complete the operation so as not to hang the processor.

Implication: The processor may hang.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R10. The Processor Signals Page-Fault Exception (#PF) Instead of Alignment Check Exception (#AC) on an Unlocked CMPXCHG8B Instruction

Problem: If a Page-Fault Exception (#PF) and Alignment Check Exception (#AC) both occur for an unlocked CMPXCHG8B instruction, then #PF will be flagged.

Implication: Software that depends on the Alignment Check Exception (#AC) before the Page-Fault Exception (#PF) will be affected since #PF is signaled in this case.

Workaround: Remove the software's dependency on #AC having precedence over #PF. Alternately, correct the page fault in the page fault handler and then restart the faulting instruction

Status: For the stepping affected, see the *Summary Tables of Changes.*

R11. FSW May Not Be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions

Problem: If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an FLDENV or FRSTOR instruction wraps around a 64-KB or 4-GB boundary and a page fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

Implication: When this erratum occurs, stale data will exist in the FSW.

Workaround: Ensure that the FPU operating environment and FPU state do not cross 64-KB or 4-GB boundaries. Alternately, ensure that the page fault handler restarts program execution at the faulting instruction after correcting the paging problem.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R12. Processor Issues Inconsistent Transaction Size Attributes for Locked Operation

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8-byte store unlock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R13. When the Processor Is in the System Management Mode (SMM), Debug Registers May Be Fully Writeable

Problem: When in System Management Mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

Implication: Reserved bit locations within DR6 and DR7 may become invalid.

Workaround: Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the values in the reserved bits are maintained.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R14. Shutdown and IERR# May Result Due to a Machine Check Exception on a Hyper-Threading Technology[1] Enabled Processor

Problem: When a Machine Check Exception (MCE) occurs due to an internal error, both logical processors on a Hyper-Threading Technology enabled processor normally vector to the MCE handler. However, if one of the logical processors is in the "Wait-for-SIPI" state, that logical processor will not have an MCE handler and will shut down and assert IERR#.

Implication: A processor with a logical processor in the "Wait-for-SIPI" state will shut down when an MCE occurs on the other thread.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R15. Processor May Hang under Certain Frequencies and 12.5% STPCLK# Duty Cycle

Problem: If a system de-asserts STPCLK# at a 12.5% duty cycle, the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

Implication: When this erratum occurs, the processor will hang.

Workaround: If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R16. System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL}

Problem: A processor internal cache fatal data ECC error may cause the processor to issue a BWL transaction to the same cache line address as an outstanding BRL or BRIL. As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

Implication: The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the System Bus under this scenario.

Workaround: System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R17. A Write to an APIC Registers Sometimes May Appear to Have Not Occurred

Problem: With respect to the retirement of instructions, stores to the uncacheable memory-based APIC register space are handled in a non-synchronized way. For example if an instruction that masks the interrupt flag, e.g. CLI, is executed soon after an uncacheable write to the Task Priority Register (TPR) that lowers the APIC priority, the interrupt masking operation may take effect before the actual priority has been lowered. This may cause interrupts whose priority is lower than the initial TPR, but higher than the final TPR, to not be serviced until the interrupt enabled flag is finally set, i.e. by STI instruction. Interrupts will remain pending and are not lost.

Implication: In this example the processor may allow interrupts to be accepted but may delay their service.

Workaround: This non-synchronization can be avoided by issuing an APIC register read after the APIC register write. This will force the store to the APIC register before any subsequent instructions are executed. No commercial operating system is known to be impacted by this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R18. Some Front Side Bus I/O Specifications Are Not Met

Problem: The following front side bus I/O specifications are not met:

- The $V_{IH(min)}$ for the GTL+ signals is specified as GTLREF + (0.10 * V_{CC}) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is specified as Vcc/2 + (0.10 * V_{CC}) [V].

Implication: This erratum can cause functional failures depending upon system bus activity. It can manifest itself as data parity, address parity, and/or machine check errors.

Workaround: Due to this erratum, the system should meet the following voltage levels and processor timings:

- The $V_{IH(min)}$ for GTL+ signals is now GTLREF + (0.20 * V_{CC}) [V].
- The $V_{IH(min)}$ for the Asynchronous GTL+ signals is now Vcc/2 + (0.20 * V_{CC}) [V].

Status: For the steppings affected, see the *Summary Tables of Changes*.

R19. Parity Error in the L1 Cache May Cause the Processor to Hang

Problem: If a locked operation accesses a line in the L1 cache that has a parity error, it is possible that the processor may hang while trying to evict the line.

Implication: If this erratum occurs, it may result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R20. BPM4# Signal Not Being Asserted According to Specification

Problem: BPM4# signal is not being asserted according to the specification. This may cause incorrect operation of In-Target Debuggers, particularly at higher FSB frequencies.

Implication: In-Target Debuggers may not function at higher than 133/533 MHz FSB.

Workaround: One method is to reduce the FSB common clock frequency to 133 MHz or lower. For higher FSB speeds, In-Target Debuggers have a built-in function (test2010) that tells the hardware to ignore BPM4# assertions. This may degrade the debugger performance but will give correct results.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R21. Sequence of Locked Operations Can Cause Two Threads to Receive Stale Data and Cause Application Hang

Problem: While going through a sequence of locked operations, it is possible for the two threads to receive stale data. This is a violation of expected memory ordering rules and causes the application to hang.

Implication: When this erratum occurs in an Hyper-Threading Technology enabled system, an application may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R22. A 16-bit Address Wrap Resulting from a Near Branch (Jump or Call) May Cause an Incorrect Address to Be Reported to the #GP Exception Handler

Problem: If a 16-bit application executes a branch instruction that causes an address wrap to a target address outside of the code segment, the address of the branch instruction should be provided to the general protection exception handler. It is possible that, as a result of this erratum, that the general protection handler may be called with the address of the branch target.

Implication: The 16-bit software environment which is affected by this erratum, will see that the address reported by the exception handler points to the target of the branch, rather than the address of the branch instruction.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R23. Bus Locks and SMC Detection May Cause the Processor to Hang Temporarily

Problem: The processor may temporarily hang in a Hyper-Threading Technology enabled system if one logical processor executes a synchronization loop that includes one or more locks and is waiting for release by the other logical processor. If the releasing logical processor is executing instructions that are within the detection range of the self -modifying code (SMC) logic, then the processor may be locked in the synchronization loop until the arrival of an interrupt or other event.

Implication: If this erratum occurs in an HT Technology enabled system, the application may temporarily stop making forward progress. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R24. PWRGOOD and TAP Signals Maximum Input Hysteresis Higher Than Specified

Problem: The maximum input hysteresis for the PWRGOOD and TAP input signals are specified at 350 mV. The actual value could be as high as 800 mV.

Implication: The PWRGOOD and TAP inputs may switch at different levels than previously documented specifications. Intel has not observed any issues in validation or simulation as a result of this erratum.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R25. Incorrect Physical Address Size Returned by CPUID Instruction

Problem: The CPUID instruction Function 80000008H (Extended Address Sizes Function) returns the address sizes supported by the processor in the EAX register. This Function returns an incorrect physical address size value of 40 bits. The correct physical address size is 36 bits.

Implication: Function 80000008H returns an incorrect physical address size value of 40 bits.

Workaround: None identified.

Status: For the steppings affected, see the Summary *Tables of Changes.*

R26. Incorrect Debug Exception (#DB) May Occur When a Data Breakpoint Is Set on an FP Instruction

Problem: The default Microcode Floating Point Event Handler routine executes a series of loads to obtain data about the FP instruction that is causing the FP event. If a data breakpoint is set on the instruction causing the FP event, the load in the microcode routine will trigger the data breakpoint resulting in a Debug Exception.

Implication: An incorrect Debug Exception (#DB) may occur if data breakpoint is placed on an FP instruction. Intel has not observed this erratum with any commercially available software or system.

Workaround: None identified.

Status: For the steppings affected, see the Summary *Tables of Changes.*

R27. xAPIC May Not Report Some Illegal Vector Errors

Problem: The local xAPIC has an Error Status Register, which records all errors. The bit 6 (the Receive Illegal Vector bit) of this register, is set when the local xAPIC detects an illegal vector in a received message. When an illegal vector error is received on the same internal clock that the error status register is being written (due to a previous error), bit 6 does not get set and illegal vector errors are not flagged

Implication: The xAPIC may not report some Illegal Vector errors when they occur at approximately the same time as other xAPIC errors. The other xAPIC errors will continue to be reported.

Workaround: None identified

Status: For the stepping affected, see the *Summary Tables of Changes.*

R28. Enabling No-Eviction Mode (NEM) May Prevent the Operation of the Second Logical Processor in a Hyper-Threading Technology Enabled Processor

Problem: In an HT Technology enabled system, when NEM is enabled by setting bit 0 of MSR 080h (IA32_BIOS_CACHE_AS_RAM), the second logical processor may fail to wake up from "Wait-for-SIPI" state.

Implication: In an HT Technology enabled system, the second logical processor may not respond to SIPI. The OS will continue to operate but with fewer logical processors than expected.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the stepping affected, see the *Summary Tables of Changes.*

R29. Incorrect Duty Cycle is Chosen when On-Demand Clock Modulation Is Enabled in a Processor Supporting Hyper-Threading Technology

Problem: When a processor supporting Hyper-Threading Technology enables On-Demand Clock Modulation on both logical processors, the processor is expected to select the lowest duty cycle of the two potentially different values. When one logical processor enters the AUTOHALT state, the duty cycle implemented should be unaffected by the halted logical processor. Due to this erratum, the duty cycle is incorrectly chosen to be the higher duty cycle of both logical processors.

Implication: Due to this erratum, higher duty cycle may be chosen when the On-Demand Clock Modulation is enabled on both logical processors.

Workaround: None identified at this time

Status: For the stepping affected, see the *Summary Tables of Changes.*

R30. Memory Aliasing of Pages As Uncacheable Memory Type and Write Back (WB) May Hang the System

Problem: When a page is being accessed as either Uncacheable (UC) or Write Combining (WC) and WB, under certain bus and memory timing conditions, the system may loop in a continual sequence of UC fetch, implicit writeback, and Request For Ownership (RFO) retries.

Implication: This erratum has not been observed in any commercially available operating system or application. The aliasing of memory regions, a condition necessary for this erratum to occur, is documented as being unsupported in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 3, section 10.12.4, Programming the PAT. However, if this erratum occurs the system may hang.

Workaround: The pages should not be mapped as either UC or WC and WB at the same time.

Status: For the stepping affected, see the *Summary Tables of Changes.*

R31. Interactions between the Instruction Translation Lookaside Buffer (ITLB) and the Instruction Streaming Buffer May Cause Unpredictable Software Behavior

Problem: Complex interactions within the instruction fetch/decode unit may make it possible for the processor to execute instructions from an internal streaming buffer containing stale or incorrect information.

Implication: When this erratum occurs, an incorrect instruction stream may be executed resulting in unpredictable software behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the stepping affected, see the *Summary Tables of Changes.*

R32. STPCLK# Signal Assertion under Certain Conditions May Cause a System Hang

Problem: The assertion of STPCLK# signal before a logical processor awakens from the "Wait-for-SIPI" state for the first time, may cause a system hang. A processor supporting Hyper-Threading Technology may fail to initialize appropriately, and may not issue a Stop Grant Acknowledge special bus cycle in response to the second STPCLK# assertion

Implication: When this erratum occurs in an HT Technology enabled system, it may cause a system hang.

Workaround: BIOS should initialize the second thread of the processor supporting Hyper-Threading Technology prior to STPCLK# assertion. Additionally, it is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R33. Missing Stop Grant Acknowledge Special Bus Cycle May Cause a System Hang

Problem: A Stop Grant Acknowledge special bus cycle being deferred by the processor for a period of time long enough for the chipset to de-assert and then re-assert STPCLK# signal may cause a system hang. A processor supporting Hyper-Threading Technology may fail to detect the de-assertion and re-assertion of STPCLK# signal, and may not issue a Stop Grant Acknowledge special bus cycle in response to the second STPCLK# assertion.

Implication: When this erratum occurs in an HT Technology enabled system, it may cause a system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R34. Changes to CR3 Register Do Not Fence Pending Instruction Page Walks

Problem: When software writes to the CR3 register, it is expected that all previous/outstanding code, data accesses and page walks are completed using the previous value in CR3 register. Due to this erratum, it is possible that a pending instruction page walk is still in progress, resulting in an access (to the PDE portion of the page table) that may be directed to an incorrect memory address.

Implication: The results of the access to the PDE will not be consumed by the processor so the return of incorrect data is benign. However, the system may hang if the access to the PDE does not complete with data (e.g. infinite number of retries).

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R35. Simultaneous Page Faults at Similar Page Offsets on Both Logical Processors of a Hyper-Threading Technology Enabled Processor May Cause Application Failure

Problem: An incorrect value of CR2 may be presented to one of the logical processors of an Hyper-Threading Technology enabled processor if a page access fault is encountered on one logical processor in the same clock cycle that the other logical processor also encounters a page fault. Both accesses must cross the same 4 byte aligned offset for this erratum to occur. Only a small percentage of such simultaneous accesses are vulnerable. The vulnerability of the alignment for any given fault is dependent on the state of other circuitry in the processor. Additionally, a third fault from an access that occurs sequentially after one of these simultaneous faults has to be pending at the time of the simultaneous faults. This erratum is caused by a one-cycle hole in the logic that controls the timing by which a logical processor is allowed to access an internal asynchronous fault address register. The end result is that the value of CR2 presented to one logical processor may be corrupted.

Implication: The operating system is likely to terminate the application that generated an incorrect value of CR2.

Workaround: An operating system or page management software can significantly reduce the already small possibility of encountering this failure by restarting or retrying the faulting instruction and only terminate the application on a subsequent failure of the same instruction. It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*

R36. The State of the Resume Flag (RF Flag) in a Task-State Segment (TSS) May Be Incorrect

Problem: After executing a JMP instruction to the next (or other) task through a hardware task switch, it is possible for the state of the RF flag (in the EFLAGS register image) to be incorrect.

Implication: The RF flag is normally used for code breakpoint management during debug of an application. It is not typically used during normal program execution. Code breakpoints or single step debug behavior in the presence of hardware task switches, therefore, may be unpredictable as a result of this erratum. This erratum has not been observed in commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R37. Using STPCLK# and Executing Code from Very Slow Memory Could Lead to a System Hang

Problem: The system may hang when the following conditions are met:

1. Periodic STPCLK# mechanism is enabled via the chipset
2. Hyper-Threading Technology is enabled
3. One logical processor is waiting for an event (i.e. hardware interrupt)
4. The other logical processor executes code from very slow memory such that every code fetch is deferred long enough for the STPCLK# to be re-asserted.

Implication: If this erratum occurs, the processor will go into and out of the sleep state without making forward progress, since the logical processor will not be able to service any pending event. This erratum has not been observed in any commercial platform running commercial software.

Workaround: None

Status: For the steppings affected, see the *Summary Tables of Changes.*

R38. Processor Provides a 4-Byte Store Unlock after an 8-Byte Load Lock

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8 byte load lock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None identified at this time.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R39. Data Breakpoints on the High Half of a Floating Point Line Split May Not Be Captured

Problem: When a floating point load which splits a 64-byte cache line gets a floating point stack fault, and a data breakpoint register maps to the high line of the floating point load, internal boundary conditions exist that may prevent the data breakpoint from being captured.

Implication: When this erratum occurs, a data breakpoint will not be captured.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R40. CPUID Instruction May Report Incorrect L2 Associativity in Leaf 0x80000006

Problem: L2 associativity reported by CPUID with EAX=80000006H instruction may be incorrect.

Implication: Software may see an incorrect L2 associativity when viewed via CPUID with EAX=80000006H, however, when viewed via CPUID with EAX=4H, the associativity value is correct.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R41. The FP_ASSIST EMON Event May Return an Incorrect Count

Problem: The performance monitoring event, FP_ASSIST, may incorrectly calculate the number of events if denormals or SSE loads are encountered.

Implication: When this erratum occurs, the FP_ASSIST event may not calculate the correct number of events. As a result, performance optimization software such as Intel VTune™ Performance Analyzers may not be able to take advantage of certain scenarios.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R42. Machine Check Exceptions May not Update Last-Exception Record MSRs (LERs)

Problem: The Last-Exception Record MSRs (LERs) may not get updated when Machine Check Exceptions occur.

Implication: When this erratum occurs, the LER may not contain information relating to the machine check exception. They will contain information relating to the exception prior to the machine check exception.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R43. MOV CR3 Performs Incorrect Reserved Bit Checking When in PAE Paging

Problem: The MOV CR3 instruction should perform reserved bit checking on the upper unimplemented address bits. This checking range should match the address width reported by CPUID instruction 0x8000008. This erratum applies whenever PAE is enabled.

Implication: Software that sets the upper address bits on a MOV CR3 instruction and expects a fault may fail. This erratum has not been observed with commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R44. Stores to Page Tables May Not Be Visible to Pagewalks for Subsequent Loads without Serializing or Invalidating the Page Table Entry

Problem: Under rare timing circumstances, a page table load on behalf of a programmatically younger memory access may not get data from a programmatically older store to the page table entry if there is not a fencing operation or page translation invalidate operation between the store and the younger memory access. Refer to the IA-32 Intel® Architecture Software Developer's Manual for the correct way to update page tables. Software that conforms to the Software Developer's Manual will operate correctly.

Implication: If the guidelines in the Software Developer's Manual are not followed, stale data may be loaded into the processor's Translation Lookaside Buffer (TLB) and used for memory operations. This erratum has not been observed with any commercially available software.

Workaround: The guidelines in the IA-32 Intel® Architecture Software Developer's Manual should be followed.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R45. Execution of IRET or INTn Instructions May Cause Unexpected System Behavior

Problem: There is a small window of time, requiring alignment of many internal micro architectural events, during which the speculative execution of the IRET or INTn instructions in protected or IA-32e mode may result in unexpected software or system behavior.

Implication: This erratum may result in unexpected instruction execution, events, interrupts or a system hang when the IRET instruction is executed. The execution of the INTn instruction may cause debug breakpoints to be missed.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R46. A Split Store Memory Access May Miss a Data Breakpoint

Problem: It is possible for a data breakpoint specified by a linear address to be missed during a split store memory access. The problem can happen with or without paging enabled.

Implication: This erratum may limit the debug capability of a debugger software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R47. EFLAGS.RF May Be Incorrectly Set after an IRET Instruction

Problem: EFLAGS.RF is used to disable code breakpoints. After an IRET instruction, EFLAGS.RF may be incorrectly set or not set depending on its value right before the IRET instruction.

Implication: A code breakpoint may be missed or an additional code breakpoint may be taken on next instruction.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R48. Read for Ownership and Simultaneous Fetch May Cause the Processor to Hang

Problem: The processor may hang when it attempts to fetch from cache line X and line X+1 simultaneously with a Read for Ownership to cache line X. If the fetch to cache line X+1 occur within a small window of time, the processor will detect this as self-modifying code and the Read for Ownership will be infinitely recycled.

Implication: If this erratum occurs, the processor may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R49. Writing the Echo TPR Disable Bit in IA32_MISC_ENABLE May Cause a #GP Fault

Problem: Writing a '1' to the Echo TPR disable bit (bit 23) in IA32_MISC_ENABLE may incorrectly cause a #GP fault.

Implication: A #GP fault may occur if the bit is set to a '1'.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R50. Cache Lock with Simultaneous Invalidate External Snoop and SMC Check May Cause the Processor to Hang

Problem: Under rare timing conditions, the processor may hang when it attempts to execute a cache lock to a cache line location while simultaneously there is a go to invalidate external snoop of the same cache line location and the processor is checking for Self-Modifying Code (SMC) of an unrelated cache line.

Implication: If this erratum occurs, the processor may hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R51. IRET Instruction Performing Task Switch May Not Serialize the Processor Execution

Problem: When an IRET instruction is executed and the NT (Nested Task) flag in the EFLAGS register is set, then buffered writes may not be drained to memory before the next instruction is fetched and executed.

Implication: Executing an IRET instruction when the NT flag in the EFLAGS register is set may not insure that all pending memory transactions have completed.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R52. Incorrect Access Controls to MSR_LASTBRANCH_0_FROM_LIP MSR Registers

Problem: When an access is made to the MSR_LASTBRANCH_0_FROM_LIP MSR register, an expected #GP fault may not happen.

Implication: A read of the MSR_LASTBRANCH_0_FROM_LIP MSR register may not cause a #GP fault.

Workaround: It is possible for the BIOS to contain a workaround for this erratum

Status: For the steppings affected, see the *Summary Tables of Changes*.

R53. Recursive Page Walks May Cause a System Hang

Problem: A page walk, accessing the same page table entry multiple times but at different levels of the page table, which causes the page table entry to have its Access bit set may result in a system hang.

Implication: When this erratum occurs, the system may experience a hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R54. WRMSR to bit[0] of IA32_MISC_ENABLE Register Changes Only One Logical Processor on a Hyper-Threading Technology Enabled Processor

Problem: On an HT enabled processor, a write to the fast-strings feature bit[0] of IA32_MISC_ENABLE register changes the setting for the current logical processor only.

Implication: Due to this erratum, the non-current logical processor may not update fast-strings feature bit[0] of IA32_MISC_ENABLE register.

Workaround: BIOS may set the fast-strings enable bit on both logical processors to workaround this erratum. It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R55. VERR/VERW Instructions May Cause #GP Fault When Descriptor Is in Non-canonical Space

Problem: If a descriptor referenced by the selector specified for the VERR or VERW instructions is in non-canonical space, it may incorrectly cause a #GP fault on a processor supporting Intel® Extended Memory 64 Technology (Intel® EM64T).

Implication: Operating systems or drivers that reference a selector in non-canonical space may experience an unexpected #GP fault. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R56. The Base of a Null Segment May Be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, the base of a null segment may be non-zero.

Implication: Due to this erratum, Intel EM64T enabled systems may encounter unexpected behavior when accessing memory using the null selector.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R57. Upper 32 Bits of FS/GS with Null Base May Not Get Cleared in Virtual-8086 Mode on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled

Problem: For processors with Intel EM64T enabled, the upper 32 bits of the FS and GS data segment registers corresponding to a null base may not get cleared when segments are loaded in Virtual-8086 mode.

Implication: This erratum may cause incorrect data to be loaded or stored to memory if FS/GS is not initialized before use in 64-bit mode. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R58. Processor May Fault When the Upper 8 Bytes of Segment Selector Is Loaded from a Far Jump through a Call Gate via the Local Descriptor Table

Problem: In IA-32e mode of the Intel EM64T processor, control transfers through a call gate via the Local Descriptor Table (LDT) that uses a 16-byte descriptor, the upper 8-byte access may wrap and access an incorrect descriptor in the LDT. This only occurs on an LDT with a LIMIT>0x10008 with a 16-byte descriptor that has a selector of 0xFFFC.

Implication: In the event this erratum occurs, the upper 8-byte access may wrap and access an incorrect descriptor within the LDT, potentially resulting in a fault or system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R59. Loading a Stack Segment with a Selector that References a Non-canonical Address Can Lead to a #SS Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T is in IA-32e mode, loading a stack segment with a selector which references a non-canonical address will result in a #SS fault instead of a #GP fault.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter unexpected behavior.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R60. FXRSTOR May Not Restore Non-canonical Effective Addresses on Processors with Intel® Extended Memory 64 Technology (Intel® EM64T) Enabled

Problem: If an x87 data instruction has been executed with a non-canonical effective address, FXSAVE may store that non-canonical FP Data Pointer (FDP) value into the save image. An FXRSTOR instruction executed with 64-bit operand size may signal a General Protection Fault (#GP) if the FDP or FP Instruction Pointer (FIP) is in non-canonical form.

Implication: When this erratum occurs, Intel EM64T enabled systems may encounter an unintended #GP fault.

Workaround: Software should avoid using non-canonical effective addressing in EM64T enabled processors. BIOS can contain a workaround for this erratum removing the unintended #GP fault on FXRSTOR.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R61. A Push of ESP That Faults May Zero the Upper 32 Bits of RSP

Problem: In the event that a push ESP instruction, that faults, is executed in compatibility mode, the processor will incorrectly zero upper 32-bits of RSP.

Implication: A Push of ESP in compatibility mode will zero the upper 32-bits of RSP. Due to this erratum, this instruction fault may change the contents of RSP. This erratum has not been observed in commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R62. Enhanced Halt State (C1E) Voltage Transition May Affect a System's Power Management in a Hyper-Threading Technology Enabled Processor

Problem: In an Hyper-Threading Technology enabled system, the second logical Processor may fail to wake up from "Wait-for-SIPI" state during a C1E voltage transition.

Implication: This erratum may affect a system's entry into the power management mode offered by the C1E event for HT Technology enabled platforms.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R63. Enhanced Halt State (C1E) May Not Be Entered in a Hyper-Threading Technology Enabled Processor

Problem: If the IA32_MISC_ENABLE MSR (0x1A0) C1E enable bit is not set prior to an INIT event on an HT Technology enabled system, the processor will not enter C1E until the next SIPI wakeup event for the second logical processor.

Implication: Due to this erratum, the processor will not enter C1E state.

Workaround: If C1E is supported in the system, the IA32_MISC_ENABLE MSR should be enabled prior to issuing the first SIPI to the second logical processor.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R64. When the Execute Disable Bit Function Is Enabled a Page-fault in a Mispredicted Branch May Result in a Page-fault Exception

Problem: If a page-fault in a mispredicted branch occurs in the ITLB, it should not be reported by the processor. However, if the execute disable bit function is enabled (IA32_EFER.NXE = 1) and there is a page-fault in a mispredicted branch in the ITLB, a page-fault exception may occur.

Implication: When this erratum occurs, a page-fault exception may occur.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R65. Execute Disable Bit Set with AD Assist May Cause Livelock

Problem: If Execute Disable Bit is set and the resulting page requires the processor to set the A and/or D bit (Access and/or Dirty bit) in the PTE, then the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R66. The Execute Disable Bit Fault May Be Reported before Other Types of Page Fault When Both Occur

Problem: If the Execute Disable Bit is enabled and both the Execute Disable Bit fault and page faults occur, the Execute Disable Bit fault will be reported prior to other types of page fault being reported.

Implication: No impact to properly written code since both types of faults will be generated but in the opposite order.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R67. Writes to IA32_MISC_ENABLE May Not Update Flags for Both Logical Processors

Problem: On processors supporting Hyper-Threading Technology with Execute Disable Bit feature, writes to IA32_MISC_ENABLE may only update IA32_EFER.NXE for the current logical processor.

Implication: Due to this erratum, the non-current logical processor may not update its IA32_EFER.NXE bit.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R68. Execute Disable Bit Set with CR4.PAE May Cause Livelock

Problem: If the Execute Disable Bit of IA32_MISC_ENABLE is set along with PAE bit of CR4 (IA32_EFER.NXE & CR4.PAE), the processor may livelock.

Implication: When this erratum occurs, the processor may livelock resulting in a system hang or operating system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R69. Checking of Page Table Base Address May Not Match the Address Bit Width Supported by the Platform

Problem: If the page table base address, included in the page map level-4 table, page-directory pointer table, page-directory table or page table, exceeds the physical address range supported by the platform (e.g. 36-bit) and it is less than the implemented address range (e.g. 40-bit), the processor does not check if the address is invalid.

Implication: If software sets such invalid physical address in those tables, the processor does not generate a page fault (#PF) upon access to that virtual address, and the access results in an incorrect read or write. If BIOS provides only valid physical address ranges to the operating system, this erratum will not occur.

Workaround: BIOS must provide valid physical address ranges to the operating system.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R70. The IA32_MCi_STATUS MSR May Improperly Indicate that Additional MCA Information May Have Been Captured

Problem: When a data parity error is detected and the bus queue is busy, the ADDRV and MISCV bits of the IA32_MCi_STATUS register may be asserted even though the contents of the IA32_MCi_ADDR and IA32_MCi_MISC MSRs were not properly captured.

Implication: If this erratum occurs, the MCA information captured in the IA32_MCi_ADDR and IA32_MCi_MISC may not correspond to the reported machine-check error, even though the ADDRV and MISCV are asserted.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R71. Execution of an Instruction with a Code Breakpoint Inhibited by the RF (Resume Flag) Bit May Be Delayed by an RFO (Request for Ownership) from Another Bus Agent

Problem: In Hyper-Threading Technology enabled parts, execution of an instruction with a code breakpoint inhibited by the RF bit may be delayed by an RFO from another bus agent. An infinite stream of these RFOs may prevent the software from making forward progress.

Implication: If this erratum occurs, the software may experience a delay in making forward progress or it may hang. Intel has not observed this erratum with any commercially available software or system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R72. With TF (Trap Flag) Asserted, FP Instruction That Triggers an Unmasked FP Exception May Take Single Step Trap before Retirement of Instruction

Problem: If an FP instruction generates an unmasked exception with the EFLAGS.TF=1, it is possible for external events to occur, including a transition to a lower power state. When resuming from the lower power state, it may be possible to take the single step trap before the execution of the original FP instruction completes.

Implication: A Single Step trap will be taken when not expected.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R73. MCA Corrected Memory Hierarchy Error Counter May Not Increment Correctly

Problem: An MCA corrected memory hierarchy error counter can report a maximum of 255 errors. Due to the incorrect increment of the counter, the number of errors reported may be incorrect.

Implication: Due to this erratum, the MCA counter may report incorrect number of soft errors.

Workaround: None Identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R74. BTS (Branch Trace Store) and PEBS (Precise Event Based Sampling) May Update Memory outside the BTS/PEBS Buffer

Problem: If the BTS/PEBS buffer is defined such that:

- The difference between BTS/PEBS buffer base and BTS/PEBS absolute maximum is not an integer multiple of the corresponding record sizes
- BTS/PEBS absolute maximum is less than a record size from the end of the virtual address space
- The record that would cross BTS/PEBS absolute maximum will also continue past the end of the virtual address space

A BTS/PEBS record can be written that will wrap at the 4G boundary (IA32) or 2^64 boundary (Intel EM64T mode), and write memory outside of the BTS/PEBS buffer.

Implication: Software that uses BTS/PEBS near the 4G boundary (IA32) or 2^64 boundary (Intel EM64T mode), and defines the buffer such that it does not hold an integer multiple of records can update memory outside the BTS/PEBS buffer.

Workaround: Define BTS/PEBS buffer such that BTS/PEBS absolute maximum minus BTS/PEBS buffer base is integer multiple of the corresponding record sizes as recommended in the IA-32 Intel® Architecture Software Developer's Manual, Volume 3.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R75. The Base of an LDT (Local Descriptor Table) Register May be Non-zero on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of an Intel EM64T-enabled processor, the base of an LDT register may be non-zero.

Implication: Due to this erratum, Intel EM64T-enabled systems may encounter unexpected behavior when accessing an LDT register using the null selector. There may be no #GP fault in response to this access.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R76. L-bit of the CS and LMA bit of the IA32_EFER Register May Have an Erroneous Value For One Instruction Following a Mode Transition in a Hyper-Threading Enabled Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In an Intel® EM64T enabled Processor, the L-bit of the Code Segment (CS) descriptor may not update with the correct value in an HT environment. This may occur in a small window when one logical processor is making a transition from compatibility mode to 64-bit mode (or vice-versa) while the other logical processor is being stalled. A similar problem may occur for the observation of the EFER.LMA bit by the decode logic.

Implication: The first instruction following a mode transition may be decoded as if it was still in the previous mode. For example, this may result in an incorrect stack size used for a stack operation, i.e. a write of only 4-bytes and an adjustment to ESP of only 4 in 64-bit mode. The problem can manifest itself, however, on any instruction which would behave differently in 64-bit mode than in compatibility mode.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R77. Memory Ordering Failure May Occur with Snoop Filtering Third Party Agents after Issuing and Completing a BWIL (Bus Write Invalidate Line) or BLW (Bus Locked Write) Transaction

Problem: Under limited circumstances, the processors may, after issuing and completing a BWIL or BLW transaction, retain data from the addressed cache line in shared state even though the specification requires complete invalidation. This data retention may also occur when a BWIL transaction's self-snooping yields HITM snoop results.

Implication: A system may suffer memory ordering failures if its central agent incorporates coherence sequencing which depends on full self-invalidation of the cache line associated (1) with BWIL and BLW transactions, or (2) all HITM snoop results without regard to the transaction type and snoop results source.

Workaround: 1. The central agent can issue a bus cycle that causes a cache line to be invalidated (Bus Read Invalidate Line (BRIL) or BWIL transaction) in response to a processor-generated BWIL (or BLW) transaction to insure complete invalidation of the associated cache line. If there are no intervening processor-originated transactions to that cache line, the central agent's invalidating snoop will get a clean snoop result.

Or

2. Snoop filtering central agents can:

a. Not use processor-originated BWIL or BLW transactions to update their snoop filter information, or

b. Update the associated cache line state information to shared state on the originating bus (rather than invalid state) in reaction to a BWIL or BLW.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R78. Control Register 2 (CR2) Can be Updated during a REP MOVS/STOS Instruction with Fast Strings Enabled

Problem: Under limited circumstances while executing a REP MOVS/STOS string instruction, with fast strings enabled, it is possible for the value in CR2 to be changed as a result of an interim paging event, normally invisible to the user. Any higher priority architectural event that arrives and is handled while the interim paging event is occurring may see the modified value of CR2.

Implication: The value in CR2 is correct at the time that an architectural page fault is signaled. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R79. TPR (Task Priority Register) Updates during Voltage Transitions of Power Management Events May Cause a System Hang

Problem: Systems with Echo TPR Disable (R/W) bit (bit [23] of IA32_MISC_ENABLE register) set to '0' (default), where xTPR messages are being transmitted on the system bus to the processor, may experience a system hang during voltage transitions caused by the power management events.

Implication: This may cause a system hang during voltage transitions of power management events.

Workaround: It is possible for the BIOS to contain a workaround for this erratum. The BIOS workaround disables the Echo TPR updates on affected steppings.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R80. REP STOS/MOVS Instructions with RCX >=2^32 May Cause a System Hang

Problem: In IA-32e mode using Intel EM64T-enabled processors, executing a repeating string instruction with the iteration count greater than or equal to 2^32 and a pending event may cause the REP STOS/MOVS instruction to live lock and hang.

Implication: When this erratum occurs, the processor may live lock and result in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: Do not use strings larger than 4 GB.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R81. An REP MOVS or an REP STOS Instruction with RCX >= 2^32 May Fail to Execute to Completion or May Write to Incorrect Memory Locations on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, an REP MOVS or an REP STOS instruction executed with the register RCX >= 2^32, may fail to execute to completion or may write data to incorrect memory locations.

Implication: This erratum may cause an incomplete instruction execution or incorrect data in the memory. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R82. An REP LODSB or an REP LODSD or an REP LODSQ Instruction with RCX >= 2^32 May Cause a System Hang on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode using Intel EM64T-enabled processors, an REP LODSB or an REP LODSD or an REP LODSQ instruction executed with the register RCX >= 2^32 may fail to complete execution causing a system hang. Additionally, there may be no #GP fault due to the non-canonical address in the RSI register.

Implication: This erratum may cause a system hang on Intel EM64T-enabled platforms. Intel has not observed this erratum with any commercially available software.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R83. A Data Access which Spans Both the Canonical and the Non-Canonical Address Space May Hang the System

Problem: If a data access causes a page split across the canonical to non-canonical address space the processor may livelock which in turn would cause a system hang.

Implication: When this erratum occurs, the processor may livelock, resulting in a system hang. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R84. Running in SMM (System Management Mode) And L1 Data Cache Adaptive Mode May Cause Unexpected System Behavior when SMRAM is Mapped to Cacheable Memory

Problem: In a Hyper-Threading Technology-enabled system, unexpected system behavior may occur if a change is made to the value of the CR3 result from an RSM (Resume From System Management) instruction while in L1 data cache adaptive mode (IA32_MISC_ENABLES MSR 0x1a0, bit 24). This behavior will only be visible when SMRAM is mapped into WB/WT cacheable memory on SMM entry and exit.

Implication: This erratum can have multiple failure symptoms including incorrect data in memory. Intel has not observed this erratum with any commercially available software.

Workaround: Disable L1 data cache adaptive mode by setting the L1 Data Cache Context Mode control (bit 24) of the IA32_MISC_ENABLES MSR (0x1a0) to 1.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R85. CPUID instruction incorrectly reports CMPXCH16B as supported

Problem: A read of the CMPXCHG16B feature flag improperly indicates that the CMPXCHG16B instruction is supported.

Implication: When a processor supporting Intel EM64T attempts to execute a CMPXCH16B instruction, the system may hang rather than #UD fault.

Workaround: It is possible for the BIOS to contain a workaround for this erratum, such that the CMPXCH16B feature flag indicates that the instruction is not supported, and the execution of the CMPXCHG16B instruction results in a #UD fault.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R86. Unaligned PDPTR (Page-Directory-Pointer) Base with 32-bit Mode PAE (Page Address Extension) Paging May Cause Processor to Hang

Problem: When the MOV to CR0, CR3 or CR4 instructions are executed in legacy PAE paging mode and software is using an unaligned PDPTR base the processor may hang or an incorrect page translation may be used.

Implication: Software that is written according to Intel's alignment specification (32-byte aligned PDPTR Base) will not encounter this erratum. Intel has not observed this erratum with commercially available software. Systems may hang or experience unpredictable behavior when this erratum occurs.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R87. FXSAVE Instruction May Result in Incorrect Data on Processors Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA-32e mode of the Intel EM64T processor, the upper 32 bits of the FDP value written out to memory by the FXSAVE instruction may be incorrect.

Implication: This erratum may cause incorrect data to be saved into the memory.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R88. Compatibility Mode STOS Instructions May Alter RSI Register Results on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T is in IA-32e mode and executes a STOS instruction in compatibility mode, it may modify the RSI register contents.

Implication: When this erratum occurs, systems may encounter unexpected behavior.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R89. LDT Descriptor Which Crosses 16 bit Boundary Access Does Not Cause a #GP Fault on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: When a processor supporting Intel EM64T in IA-32e mode accesses an LDT entry (16-byte) that crosses the 0xffff limit, a #GP fault is not signaled and instead the upper 8-bytes of the entry is fetched from the wrapped around address (usually 0x0). This will cause the erroneous data to be loaded into the upper 8-bytes of the descriptor.

Implication: When this erratum occurs, systems may encounter unexpected behavior. Intel has not observed this erratum with any commercially available software.

Workaround: Software should prevent LDT selector accesses from crossing the 0xffff limit.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R90. Upper Reserved Bits are Incorrectly Checked While Loading PDPTR's on a Processor Supporting Intel® Extended Memory 64 Technology (Intel® EM64T)

Problem: In IA32 & IA-32e mode of the Intel® processor, upper reserved bits are incorrectly checked while loading PDPTR's, allowing software to set the reserved bits.

Implication: Operating system software is able to set the reserved bits which may result in an unexpected system behavior.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R91. A 64-Bit Value of Linear Instruction Pointer (LIP) May be Reported Incorrectly in the Branch Trace Store (BTS) Memory Record or in the Precise Event Based Sampling (PEBS) Memory Record

Problem: On a processor supporting Intel® EM64T,

- If an instruction fetch wraps around the 4G boundary in Compatibility Mode, the 64-bit value of LIP in the BTS memory record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).
- If a PEBS event occurs on an instruction whose last byte is at memory location FFFFFFFFh, the 64-bit value of LIP in the PEBS record will be incorrect (upper 32 bits will be set to FFFFFFFFh when they should be 0).

Implication: Intel has not observed this erratum on any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R92. It is Possible That Two specific Invalid Opcodes May Cause Unexpected Memory Accesses

Problem: A processor is expected to respond with an undefined opcode (#UD) fault when executing either opcode 0F 78 or a Grp 6 Opcode with bits 5:3 of the Mod/RM field set to 6, however the processor may respond instead, with a load to an incorrect address.

Implication: This erratum may cause unpredictable system behavior or system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R93. At Core-to-bus Ratios of 16:1 and Above Defer Reply Transactions with Non-zero REQb Values May Cause a Front Side Bus Stall

Problem: Certain processors are likely to hang the Front Side Bus (FSB) if the following conditions are met:

1. A Defer Reply transaction has a REQb[2:0] value of either 010b, 011b, 100b, 110b, or 111b, and

2. The operating bus ratio is 16:1 or higher.

When these conditions are met, the processor may incorrectly and indefinitely assert a snoop stall for the Defer Reply transaction. Such an event will block further progress on the FSB.

Implication: If this erratum occurs, the system may hang. Intel has not observed this erratum with any commercially available system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R94. The Processor May Issue Front Side Bus Transactions up to 6 Clocks after RESET# is Asserted

Problem: The processor may issue transactions beyond the documented 3 Front Side Bus (FSB) clocks and up to 6 FSB clocks after RESET# is asserted in the case of a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may issue transactions up to 6 FSB clocks after the RESET# is asserted

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R95. Front Side Bus Machine Checks May be Reported as a Result of On-Going Transactions during Warm Reset

Problem: Processor Front Side Bus (FSB) protocol/signal integrity machine checks may be reported if the transactions are initiated or in-progress during a warm reset. A warm reset is where the chipset asserts RESET# when the system is running.

Implication: The processor may log FSB protocol/signal integrity machine checks if transactions are allowed to occur during RESET# assertions.

Workaround: BIOS may clear FSB protocol/signal integrity machine checks for systems/chipsets which do not block new transactions during RESET# assertions.

Status: For the steppings affected, see the *Summary Tables of Changes*

R96. CPUID Feature Flag Reports LAHF/SAHF as Unavailable however the Execution of LAHF/SAHF May Not Result in an Invalid Opcode Exception

Problem: As described in the IA-32 Intel® Architecture Software Developer's Manual, support for LAHF/SAHF instructions in 64-bit mode has been added to Intel EM64T processors. The CPUID feature flag may indicate that the LAHF/SAHF instructions are unavailable in 64-bit mode, even though the instructions are supported and able to be executed without an Invalid Opcode exception.

Implication: The CPUID Feature Flag incorrectly reports LAHF/SAHF instructions as unavailable in 64-bit mode; however they can be executed normally.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R97. The Processor May Issue Multiple Code Fetches to the Same Cache Line for Systems with Slow Memory

Problem: Systems with long latencies on returning code fetch data from memory e.g. BIOS ROM, may cause the processor to issue multiple fetches to the same cache line, once per each instruction executed.

Implication: This erratum may slow down system boot time. Intel has not observed a failure, as a result of this erratum, in a commercially available system.

Workaround: It is possible for the BIOS to contain a workaround for this erratum for some steppings of the processor.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R98. Writing the Local Vector Table (LVT) when an Interrupt is Pending May Cause an Unexpected Interrupt

Problem: If a local interrupt is pending when the LVT entry is written, an interrupt may be taken on the new interrupt vector even if the mask bit is set.

Implication: An interrupt may immediately be generated with the new vector when a LVT entry is written, even if the new LVT entry has the mask bit set. If there is no Interrupt Service Routine (ISR) set up for that vector the system will GP fault. If the ISR does not do an End of Interrupt (EOI) the bit for the vector will be left set in the in-service register and mask all interrupts at the same or lower priority.

Workaround: Any vector programmed into an LVT entry must have an ISR associated with it, even if that vector was programmed as masked. This ISR routine must do an EOI to clear any unexpected interrupts that may occur. The ISR associated with the spurious vector does not generate an EOI, therefore the spurious vector should not be used when writing the LVT.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R99. Access to an Unsupported Address Range in Uniprocessor (UP) or Dual-processor (DP) Systems Supporting Intel® Virtualization Technology May Not Trigger Appropriate Actions

Problem: When using processors supporting Intel® Virtualization Technology and configured as dual- or single-processor-capable (i.e. not multiprocessor-capable), the processor should perform address checks using a maximum physical address width of 36. Instead, these processors will perform address checks using a maximum physical address width of 40.

Implication: Due to this erratum, actions which are normally taken upon detection of an unsupported address may not occur. Software which does not attempt to access unsupported addresses will not experience this erratum.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R100. VM Exit Due to a MOV from CR8 May Cause an Unexpected Memory Access

Problem: In a system supporting Intel® Virtualization Technology and Intel® Extended Memory 64 Technology, if the "CR8-store exiting" bit in the processor-based VM-execution control field is set and the "use TPR shadow" bit is not set, a MOV from CR8 instruction executed by a Virtual Machine Extensions (VMX) guest that causes a VM exit may generate an unexpected memory access.

Implication: When this erratum occurs, a read access to unexpected address may be issued to the chipset. Subsequent side effects are dependent on chipset operation and may include system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R101. The Processor May Incorrectly Respond to Machine Checks during VM Entry/Exit Transitions

Problem: In systems supporting Intel® Virtualization Technology, when machine checks are encountered during VM entry/exit transitions, the processor is expected to respond with a VM exit (if a machine check occurs during VM entry) or abort (if a machine check occurs during VM exit). As a result of this erratum when machine checks occur during VM entry/exit transitions the processor will attempt to service the machine check which may lead to IERR-shutdown or execution of the Machine Check handler, dependent on the CR4.MCE setting.

Implication: The system may end up in the shutdown state if CR4.MCE is not set.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R102. INIT during String Operations in the Virtual-Machine Extension (VMX) Guest Mode May Cause Unexpected System Behavior

Problem: In a system supporting Intel® Virtualization Technology, if INIT occurs during REP LODS/MOVS/STOS/INS/OUTS while the processor is executing in VMX guest mode, after servicing the INIT, the host will resume at the next instruction and does not complete the remainder of string operation.

Implication: This erratum may cause unexpected system behavior to occur.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R103. Power Down Requests May not be Serviced if a Power Down Transition is Interrupted by an In-Target Probe Event in the Presence of a Specific Type of VM Exit

Problem: In a system supporting Intel® Virtualization Technology, the processor may service a pended VM exit prior to completely exiting out of a low power state when the following sequences of events occur:

- Chip-wide power down transition occurs and
- VM exit due to a VMLaunch, VMResume, STI, POPF, POPFD, or IRET instruction is pended and
- Chip-wide power down transition is interrupted by an In-Target Probe event.

Implication: Due to this erratum the processor may not recognize further STPCLK# assertions, TM1, TM2, or Enhanced Intel SpeedStep® Technology. Intel has not observed this erratum with any commercially available software.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R104. VM EXIT Due to TPR shadow Below Threshold May Improperly Set and Cause "Blocking by STI" actions

Problem: In a system supporting Intel® Virtualization Technology and Intel® EM64T, the “blocking by STI” bit of the interruptibility-state field may be saved as 1 rather than 0. This erratum may occur when a STI instruction is executed directly prior to a MOV to CR8 which results in a VM exit due to a reduction of the TPR shadow value below the TPR threshold.

Implication: When this erratum occurs, delivery of an interrupt may be delayed by one instruction.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R105. VM Entry/Exit Writes to LSTAR/SYSCALL_FLAG MSR's May Cause Incorrect Data to be Written to Bits [63:32]

Problem: Incorrect MSR data in bits [63:32] may be observed in the following two cases;

1. When ECX contains 0xC0000084 and a VM entry/exit writes the IA32_CR_LSTAR MSR (MSR Address 0xC0000082) bits [63:32] of the data may be zeroed.

2. When ECX does not contain 0xC0000084 and a VM entry/exit writes the IA32_CR_SYSCALL_FLAG_MASK MSR (MSR Address 0xC0000084) bits [63:32] of the data may not be zeroed.

Implication: Bits [63:32] of the affected MSRs may contain the wrong data after a VM exit/entry which loads the affected MSR.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R106. Machine Check Architecture Multiple Data Parity Errors May be Reported

Problem: When the processor detects a Front Side Bus (FSB) data parity error and the core is being clocked at a ratio 12:1 or 13:1 with respect to the FSB clock, it may report multiple data parity errors. When this situation occurs, there could be multiple assertions of MCERR# until the processor receives FSB data without a parity error or a system bus reset occurs.

Implication: Multiple (extraneous) data parity errors may be reported in the system.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R107. Attempting to Use an LDT Entry when the LDTR Has Been Loaded with an Unusable Segment May Cause Unexpected Memory Accesses

Problem: In a system supporting Intel® EM64T and Intel® Virtualization Technology when the following occur,

- The LDTR is loaded during VM entry with the segment unusable bit set for the LDTR in the VMCS (Virtual-Machine Control Structure)
- The segment limit is non-zero
- The granularity bit is set.

References to a segment located in the LDT in 64-bit mode at any time later may cause the processor to exhibit unexpected behavior.

Implication: This erratum may cause unexpected memory accesses.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R108. The Execution of a VMPTRLD Instruction May Cause an Unexpected Memory Access

Problem: In a system supporting Intel® Virtualization Technology, executing VMPTRLD may cause a memory access to an address not referenced by the memory operand.

Implication: This erratum may cause unpredictable system behavior including system hang.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R109. The Execution of VMPTRLD or VMREAD May Cause an Unexpected Memory Access

Problem: On processors supporting Intel® Virtualization Technology, executing a VMPTRLD or a VMREAD instruction outside of VMX mode may result in a load to an unexpected address.

Implication: This erratum may cause a load to an unexpected memory address.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R110. IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception

Problem: In IA-32e mode, it is possible to get an Alignment Check Exception (#AC) on the IRET instruction even though alignment checks were disabled at the start of the IRET. This can only occur if the IRET instruction is returning from CPL3 code to CPL3 code. IRETs from CPL0/1/2 are not affected. This erratum can occur if the EFLAGS value on the stack has the AC flag set, and the interrupt handler's stack is misaligned. In IA-32e mode, RSP is aligned to a 16-byte boundary before pushing the stack frame.

Implication: In IA-32e mode, under the conditions given above, an IRET can get a #AC even if alignment checks are disabled at the start of the IRET. This erratum can only be observed with a software generated stack frame.

Workaround: Software should not generate misaligned stack frames for use with IRET.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R111. FS/GS Base MSRs can be Loaded from MSR-Load Areas during VM Entry or VM Exit

Problem: If the VM Exit or VM Entry MSR load area contains references to the FS or GS Base MSRs, the VM Exit and VM Entry transitions should fail. Instead, the operation will load the MSRs with the value in the corresponding MSR-load area entry.

Implication: VM Entries and VM Exits that should fail will complete successfully in this situation. If a VM entry is to virtual-8086 mode, the base address for FS or for GS may be loaded with a value that is not consistent with that mode. Intel has not observed this erratum with any commercially available software or systems.

Workaround: Software should not enter values in the MSR-load areas that correspond to either the FS Base MSR or the GS Base MSR. Software can establish the value of these registers on VM entry using the guest-state area of the Virtual-Machine Control Structure (VMCS) and on VM exit using the host-state area of the VMCS.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R112. NMI-blocking Information Recorded in VMCS May be Incorrect after a #GP on an IRET Instruction

Problem: In a system supporting Intel® Virtualization Technology, the NMI blocking bit in the Interruption-Information Field in the guest VMCS may be set incorrectly. This erratum will happen if a VMExit occurs for a #GP fault on an IRET instruction due to an EIP that violates the segment limit or is non-canonical.

Implication: If this erratum occurs, monitor software may not be able to handle #GP and then inject an NMI since monitor software does not have information about whether NMIs are blocked in the guest.

Workaround: Monitor software can workaround this bug by avoiding injection of NMI after #GP emulation.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R113. VMLAUNCH/VMRESUME May Not Fail when VMCS is Programmed to Cause VM Exit to Return to a Different Mode

Problem: VMLAUNCH/VMRESUME instructions may not fail if the value of the “host address-space size” VM-exit control differs from the setting of IA32_EFER.LMA.

Implication: Programming the VMCS to allow the monitor to be in different modes prior to VMLAUNCH/VMRESUME and after VM-exit may result in undefined behavior.

Workaround: Software should ensure that "host address-space size" VM-exit control has the same value as IA32_EFER.LMA at the time of VMLAUNCH/VMRESUME.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R114. Upper 32 bits of 'From' Address Reported through LBR or LER MSRs, BTMs or BTSs May be Incorrect

Problem: When a far transfer switches the processor from IA-32e mode to 32-bit mode, the upper 32 bits of the 'From' (source) addresses reported through the LBR (Last Branch) or LER (Last Exception Record) MSRs (Model-Specific Registers), BTMs (Branch Trace Messages) or BTSs (Branch Trace Stores) may be incorrect.

Implication: The upper 32 bits of the 'From' address debug information reported through LBR or LER MSRs, BTMs or BTSs may be incorrect.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R115. VMEntry from 64-bit Host to 32-bit Guest may Cause IERR# with Hyper-Threading Enabled

Problem: When transitioning from a 64-bit host environment to a 32-bit guest environment via a VMEntry, internal conditions in a processor with Hyper-Threading enabled may cause a speculative page-table walk to be prematurely terminated, resulting in a processor hang and the assertion of IERR#.

Implication: An IERR# may occur on VMEntry from a 64-bit to a 32-bit environment with Hyper-Threading enabled.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R116. L2 Cache ECC Machine Check Errors May be erroneously Reported after an Asynchronous RESET# Assertion

Problem: Machine check status MSRs may incorrectly report the following L2 Cache ECC machine-check errors when cache transactions are in-flight and RESET# is asserted:

•Instruction Fetch Errors (IA32_MC2_STATUS with MCA error code 153)

•L2 Data Write Errors (IA32_MC1_STATUS with MCA error code 145)

Implication: Uncorrected or corrected L2 ECC machine check errors may be erroneously reported. Intel has not observed this erratum on any commercially available system.

Workaround: When a real run-time L2 Cache ECC Machine Check occurs, a corresponding valid error will normally be logged in the IA32_MC0_STATUS register. BIOS may clear IA32_MC2_STATUS and/or IA32_MC1_STATUS for these specific errors when IA32_MC0_STATUS does not have its VAL flag set.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R117. VMExit after MOV SS and a Waiting x87 Instruction May not Clear the Interruptibility State in the VMM's Working VMCS

Problem: When Guest software executes a waiting x87 instruction, after an interrupt window which was closed due to blocking by MOVSS, a VM exit may occur due to Interrupt Window exiting = 1 in the Processor-Based VM-Execution Controls of the Controlling VMCS. This causes blocking by MOVSS bit in Interruptibility State to be incorrectly set in the VMM's Working VMCS.

Implication: If a VM Exit occurs due to the Guest Software executing a waiting x87 instruction after a MOVSS and the VMM injects a hardware interrupt or an NMI to the Guest without explicitly clearing the Blocking by MOVSS bit in the Interruptibility State field in the working VMCS, then, the VMM will see a spurious VM Entry failure.

Workaround: The VM Monitor should explicitly clear the Blocking by MOVSS bit in the Interruptibility State field in the working VMCS, before injecting a hardware interrupt or an NMI to the Guest software.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R118. VMCALL to Activate Dual-monitor Treatment of SMIs and SMM Ignores Reserved Bit settings in VM-exit Control Field

Problem: Processors supporting Intel® Virtualization Technology can execute VMCALL from within the Virtual Machine Monitor (VMM) to activate dual-monitor treatment of SMIs and SMM. Due to this erratum, if reserved bits are set to values inconsistent with VMX Capability MSRs, VMCALL may not VMFail.

Implication: VMCALL executed to activate dual-monitor treatment of SMIs and SMM may not VMFail due to incorrect reserved bit settings in VM-Exit control field.

Workaround: Software should ensure that all VMCS reserved bits are set to values consistent with VMX Capability MSRs.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R119. Using 2M/4M Pages When A20M# Is Asserted May Result in Incorrect Address Translations

Problem: An external A20M# pin if enabled forces address bit 20 to be masked (forced to zero) to emulates real-address mode address wraparound at 1 megabyte. However, if all of the following conditions are met, address bit 20 may not be masked:

- paging is enabled
- a linear address has bit 20 set
- the address references a large page
- A20M# is enabled

Implication: When A20M# is enabled and an address references a large page the resulting translated physical address may be incorrect. This erratum has not been observed with any commercially available operating system.

Workaround: Operating systems should not allow A20M# to be enabled if the masking of address bit 20 could be applied to an address that references a large page. A20M# is normally only used with the first megabyte of memory.

Status: For the steppings affected, see the *Summary Tables of Changes*.

R120. Writing Shared Unaligned Data that Crosses a Cache Line without Proper Semaphores or Barriers May Expose a Memory Ordering Issue

Problem: Software which is written so that multiple agents can modify the same shared unaligned memory location at the same time may experience a memory ordering issue if multiple loads access this shared data shortly thereafter. Exposure to this problem requires the use of a data write which spans a cache line boundary.

Implication: This erratum may cause loads to be observed out of order. Intel has not observed this erratum with any commercially available software or system.

Workaround: Software should ensure at least one of the following is true when modifying shared data by multiple agents:

- The shared data is aligned
- Proper semaphores or barriers are used in order to prevent concurrent data accesses

Status: For the steppings affected, see the *Summary Tables of Changes*.

R121. The IA32_MC0_STATUS and IA32_MC1_STATUS Overflow Bit is not set when Multiple Un-correctable Machine Check Errors Occur at the Same Time

Problem: When two enabled MC0/MC1 un-correctable machine check errors are detected in the same bank in the same internal clock cycle, the highest priority error will be logged in IA32_MC0_STATUS / IA32_MC1_STATUS register, but the overflow bit may not be set.

Implication: The highest priority error will be logged and signaled if enabled, but the overflow bit in the IA32_MC0_STATUS/ IA32_MC1_STATUS register may not be set.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

R122. Debug Status Register (DR6) Breakpoint Condition Detected Flags May be set Incorrectly

Problem: The Debug Status Register (DR6) may report detection of a spurious breakpoint condition under certain boundary conditions when either:

- A "MOV SS" or "POP SS" instruction is immediately followed by a hardware debugger breakpoint instruction, or
- Any debug register access ("MOV DRx, r32" or "MOV r32, DRx") results in a general-detect exception condition.

Implication: Due to this erratum the breakpoint condition detected flags may be set incorrectly.

Workaround: None identified.

Status: For the steppings affected, see the *Summary Tables of Changes.*

§

Specification Changes

The Specification Changes listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor 670, 660, 650, 640, and 630Δ and Intel® Pentium® 4 Processor Extreme Edition Datasheet*
- *Intel® Pentium ® 4 Processors 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521Δ Supporting Hyper-Threading Technology[1] Datasheet*

All Specification Changes will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

R1. Land Assignment Specification Change

The following lands have been reassigned as follows in the *Intel® Pentium® 4 Processor 670, 660, 650, 640, and 630Δ and Intel® Pentium® 4 Processor Extreme Edition, Intel Pentium 4 Processor Supporting Hyper-Threading Technology, Intel Pentium 4 Processor 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521Δ Datasheet: On 90 nm Process in the 775-Land LGA Package Supporting Hyper-Threading Technology*. These reassignments have been made to facilitate compatibility with future processors. These changes apply to the Land Listing and Signal Descriptions chapter in each of the documents shown above.

Y3	from RESERVED to FC17
AE3	from RESERVED to FC18
B13	from RESERVED to FC19
E5	from RESERVED to FC20
AK6	from RESERVED to FC8
D23	from RESERVED to FC9
F6	from RESERVED to FC21
J3	from RESERVED to FC22

Δ Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Over time processor numbers will increment based on changes in clock, speed, cache, FSB, or other features, and increments are not intended to represent proportional or quantitative increases in any particular feature. Current roadmap processor number progression is not necessarily representative of future roadmaps. See www.intel.com/products/processor_number for details.

§

Specification Clarifications

The Specification Clarifications listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor 670, 660, 650, 640, and 630[4] and Intel® Pentium® 4 Processor Extreme Edition Datasheet*
- *Intel® Pentium® 4 Processors 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521[4] Supporting Hyper-Threading Technology[1] Datasheet*

All Specification Clarifications will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

§

Documentation Changes

The Documentation Changes listed in this section apply to the following documents:

- *Intel® Pentium® 4 Processor 670, 660, 650, 640, and 630[Δ] and Intel® Pentium® 4 Processor Extreme Edition Datasheet*
- *Intel® Pentium® 4 Processors 570/571, 560/561, 550/551, 540/541, 530/531 and 520/521[Δ] Supporting Hyper-Threading Technology[1] Datasheet*

All Documentation Changes will be incorporated into a future version of the appropriate Pentium 4 processor documentation.

Note: Documentation changes for IA-32 Intel® Architecture Software Developer's Manual volumes 1, 2A, 2B, 3A, and 3B will be posted in a separate document IA-32 Intel® Architecture Software Developer's Manual Documentation Changes. Follow the link below to become familiar with this file.

http://developer.intel.com/design/pentium4/specupdt/252046.htm

There are no documentation changes in this Specification Update revision.

§